COOPERATIVE SELF-LOCALIZATION IN A MULTI-ROBOT-NO-LANDMARK SCENARIO USING FUZZY LOGIC

A Thesis

by

DHIRENDRA KUMAR SINHA

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

December 2004

Major Subject: Mechanical Engineering

COOPERATIVE SELF-LOCALIZATION IN A MULTI-ROBOT-NO-LANDMARK SCENARIO USING FUZZY LOGIC

A Thesis

by

DHIRENDRA KUMAR SINHA

Submitted to Texas A&M University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Approved as to style and content by:

Reza Langari (Chair of Committee)	Alexander Parlos (Member)
Mehrdad Ehsani (Member)	Dennis O'Neal (Head of the Department)

December 2004

Major Subject: Mechanical Engineering

ABSTRACT

Cooperative Self-Localization in a Multi-Robot-No-Landmark Scenario Using Fuzzy Logic. (December 2004)

Dhirendra Kumar Sinha, B.Tech., Indian Institute of Technology, Guwahati

Chair of Advisory Committee: Dr. Reza Langari

In this thesis, we develop a method using fuzzy logic to do cooperative localization. In a group of robots, at a given instant, each robot gives crisp pose estimates for all the other robots. These crisp pose values are converted to fuzzy membership functions based on various physical factors like acceleration of the robot and distance of separation of the two robots. For a given robot, all these fuzzy estimates are taken and fused together using fuzzy fusion techniques to calculate a possibility distribution function of the pose values. Finally, these possibility distributions are defuzzified using fuzzy techniques to find a crisp pose value for each robot. A MATLAB code is written to simulate this fuzzy logic algorithm. A Kalman filter approach is also implemented and then the results are compared qualitatively and quantitatively.

To my parents, brothers and wife and all the teachers and mentors who have shaped my thought process.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my thesis advisor, Dr. Reza Langari for his invaluable support, encouragement and guidance during my course of study at Texas A&M University.

I would also like to thank Dr. Alexander Parlos and Dr. Mehrdad Ehsani for serving on my thesis committee.

I sincerely appreciate the encouragement and help from Dr. Ricardo Gutierrez-Osuna and Dr. Sooyong Lee, who inspired me a lot to work in the area of robotics.

I am also very grateful to my friends and colleagues at Texas A&M University for their support and encouragement. Special thanks to Karthik Aruru and Ananth Eyunni for proofreading this thesis.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

CHAPTER I

INTRODUCTION

Introduction

In the near future, manual work will become more automated as technology improves. Robots would be employed extensively in industries, homes, human-unsafe environments like nuclear power plants, underwater explorations and space explorations. These robots need to be autonomous to really work in an efficient and reliable way. One of the very important tasks of an autonomous robot is to navigate in a given environment. Automatic navigation requires that a robot should be able to localize itself. In other words, it should know what its pose (position and orientation) is. Humans and animals determine their approximate positions from visual information and knowledge of their previous movements. For humans and animals, generally, it is sufficient to find their locations approximately. When needed, humans can always use their sophisticated wide variety of sensors to do precise localization. It is difficult to give these skills to robots because of the limitations imposed by sensor performance, computational cost and environment models.

This thesis follows the style of *IEEE Transactions on Robotics and Automation.*

A number of simple techniques of localization have been proposed based on local information about the robot itself and its surroundings. A typical technique is dead reckoning, using which mobile robots with wheels identify their current position from the rotational speed of the wheels [1]. Dead reckoning is simple and therefore easy to implement. The position given by dead reckoning is, however, influenced by the wheel-tire contact with the ground and so there are errors (odometry errors) due to slippages between the ground and wheels. These odometry errors render it impossible for any robot to follow a given trajectory sufficiently accurately.

There are many tasks that can be performed in a more efficient and robust manner using multiple robots [2]. There are many advantages of using several small moderately capable robots instead of using one large highly sophisticated robot [3]. Understandably, the reliability of such a multi-robot system is much higher than single-robot systems, enabling the team to accomplish the intended mission goals even if one member of the team fails. Although, the complexity increases in the case of multi-robot localization, the presence of multiple robots, actually, gives an advantage towards finding the pose of each robot. To this end, there has been much work done in the collaborative and cooperative localization [4]−[9]. Each robot can give pose estimates for all other robots. For each robot, the pose estimates given by all the other robots can be combined together and a final pose estimate can be calculated. Combining the information from all the robots will result in a single estimate with increased accuracy and reduced uncertainty. The advantages stemming from the exchange of information among the members of a group are crucial in the case of heterogeneous robotic colonies.

When a team is composed of different robots carrying different sensors and thus having different capabilities for self localization, the quality of the localization estimates will vary significantly across the individual group.

As discussed earlier, the pose estimates may contain errors due to wheel slippages. The uncertainty or unreliability of these pose estimates given by robots may depend upon several physical parameters which can easily be measured. But an exhaustive list of parameters and a mathematical formulation of the dependencies of pose estimates on these factors is generally not available. Therefore, there is a need to develop a model which takes these uncertainties into account. One way to incorporate the uncertainty of the pose estimates is to model the pose values as Gaussian distributions. Another way to incorporate this uncertainty is to construct fuzzy membership functions. In cooperative localization, we combine the pose estimates given by all the other robots to find the pose of one robot. If this fusion is not done carefully, it may result in degradation of the final pose.

This work describes a method for localizing the members of a mobile robot team, using the robot themselves as landmarks. That is, we describe a method using which each robot can determine the relative range, bearing and orientation of every other robot in the team, without the use of GPS, external landmarks, or instrumentation of the environment. The major factors affecting the uncertainty of the pose estimation are identified and studied. Here, the uncertain estimates are represented as “fuzzy sets” and combined to compute a final pose value for a robot.

Real time practical examples of multi-robot scenario

Multiple robots are becoming very popular and advantageous in home, industry and military areas. Some of real time examples of the use of multiple robots are as follows:

1. *Guiding human visitors*: Multiple robots are being used to guide humans in a large indoor space like offices, exhibition centers and museums. Multiple robots communicate with one another and perform assigned tasks collaboratively to reduce the over all cost and increase efficiency [10].
2. *Security and automated inventory assessment*: MDARS program, a joint Army-Navy effort is developing a robotic security and automated inventory assessment capability for use in the Department of Defense warehouses and storage sites. The program is managed by the US Army Physical Security Equipment Management Office, Ft. Belvoir, VA, with NCCOSC providing all technical direction and systems integration functions [11].
3. *Air, surface and subsurface vehicles for exploration of the planets:* At Jet Propulsion Laboratory, NASA, researchers are working on the next generation of air, surface and subsurface vehicles (lightweight, intelligent and can work without an operator at the wheel) for exploration of the planetary bodies including Mars, Venus, Jupiter's moon Europa and Saturn's largest moon Titan [12].
4. *Search and rescue operations:* National Science Foundation is putting $2.6 million into a five-year effort to turn multiple wireless robots into an

emergency search-and-rescue team. The program envisions coordinating multiple robots to carry out emergency workers' complex, high-level commands such as "search this site for survivors" or "draw a map showing which walls are collapsed" [13].

5. *Battlefield robots:* SARGE (Surveillance And Reconnaissance Ground Equipment), a battlefield robot that could reduce risk to soldiers by performing some of their more dangerous tasks, was developed at Sandia National Laboratories, *Lockheed Martin Corporation,* primarily to engage in remote surveillance [14].
6. *Lawn mower robots:* An industrial-grade robotic mower from Carnegie Mellon University is trimming golf-course fairways and greens, as well as the training field for the Pittsburgh Steelers football team. Golf-course owners who use robots to cut grass at night will be able to reduce labor costs and accommodate more players on their courses during the day [15].
7. *Collective construction by multiple robots:* study of the problem of construction by autonomous mobile robots focusing on the coordination strategy employed by the robots to solve a simple construction problem efficiently [16].

All the above practical scenarios require cooperation between various robots and thus there is a strong need of cooperative localization techniques to be developed.

Keywords and their explanations

There are some basic keywords which would be used extensively in this work as discussed below:

1. *Pose:* Pose P(k) (x_i, y_i, θ_i) represents the position and orientation coordinate values of Robot Ri with respect to the global coordinates at instant k as shown in Fig. 1. Here, x_i and y_i are the x and y coordinates of the robot with respect to the global coordinate system and θ_i is the angle of x_i with respect to the global x coordinate axis. ρ_{12} is the distance vector from R2 to R1 with respect to R2's coordinate system.

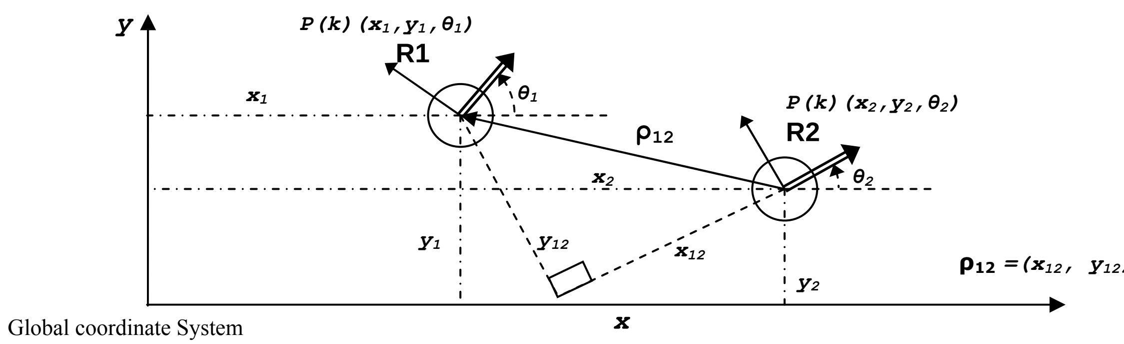


Fig. 1. Pose and range vector. The figure shows the top view of the robots R1 and R2 at instant k. Each robot has a coordinate system attached to it.

2. *Proprioceptive sensors:* The sensors which are mounted on a robot and are used to find changes in its pose are called Proprioceptive sensors, for example, see the optical wheel encoder as shown in Fig. 2.

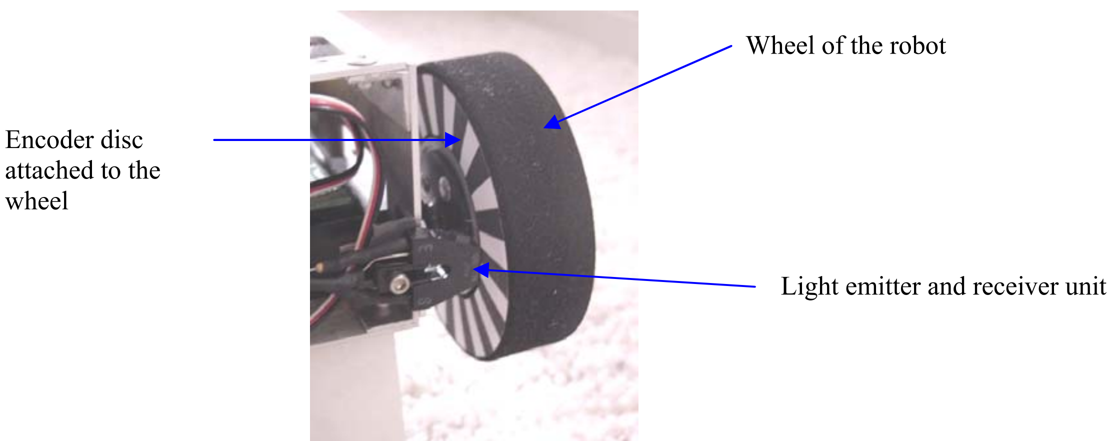


Fig. 2. Proprioceptive sensor. The optical wheel encoder disc is glued to the wheel. The light emitter continuously emits light and receiver unit receives high or low inputs based on whether the light falls on the white or black strip.

3. *Exteroceptive sensors:* The sensors which are mounted on a robot and are used to find the distance vector (magnitude and direction) to another robot are called Exteroceptive sensors, for example, omni-directional stereo camera as shown in Fig. 3.

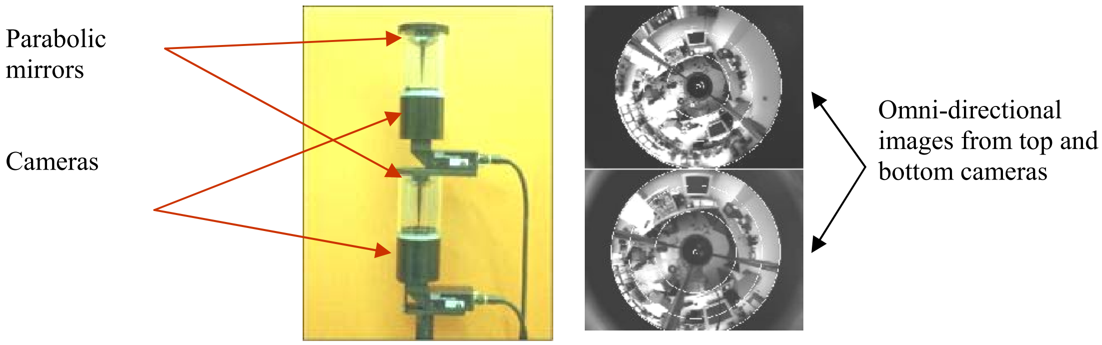


Fig. 3. Exteroceptive sensor. An example: the omni-directional stereo camera setup [17]. The two omni-directional cameras take images and then based on the pixel location for a given point, the range distance to that point can be found by simple mathematical formula.

4. *Localization:* The method of finding the pose of a particular robot at a particular instant is called localization. This is a very important problem in autonomous navigation of robots. If the robot doesn't know where it is relative to the environment, it is difficult to decide what it should do and where should it go. The robot will most likely need to have an idea of where it is to operate and act successfully.

5. *Cooperative localization:* The localization method combining the pose estimates provided by other robots in the group to find the pose for a particular robot is called as cooperative localization. The robots can cooperate with each other to help each other find the pose values.

Prior work

A number of localization techniques have been proposed in the literature. The dead reckoning method discussed in [1], [18], [19], [20], [21] identifies robot positions by calculating the amount of travel from the starting point. It does this by integrating rotations of the right and the left wheels. The dead reckoning method, however, has a serious problem. Wheel slippage causes measurement errors, which accumulates as the vehicle travels. Kato *et. al.* propose the localization in multi-robot scenario using omni directional vision cameras [22]. Using the omni-directional cameras, the range vectors to other robots can be found easily. Another positioning and localization technique is using landmarks [23]−[25]. The landmark method uses optical or other sensors installed in the robot to detect walls, pillars and other objects in the environment and also some artificially placed landmarks. The landmark method can give highly accurate positioning when the robot travels long distances, but requires the placing of landmarks. It cannot, for example, be used for planetary exploration robots, which work in uncharted environments.

Cooperative localization without any external landmarks or GPS is dealt in [26], [4], [27], [28]. Rekleitis *et. al.* analyze the advantages of cooperative robots versus a single one and discuss how, using multiple robots, the odometry errors be minimized [3]. The assumption in this work is that at any time only one robot moves and all the other are stationary and observe its motion. Concept of "portable landmarks" was introduced by Kurazume *et. al.*[29]. A group of robots is divided into two teams in order to perform cooperative positioning. At each time instant, one team is in motion while the other remains stationary and acts as landmark. In the next phase the roles are reversed until both teams reach the target.

Cooperative localization is also studied in the wireless network field [30]. Networked sensors can collaborate and aggregate large amount of sensed data to provide continuous and spatially dense observations in environmental systems such as a sea. Instrumenting the physical world, particularly for such applications, requires that the devices we use as sensor nodes be small, light, unobtrusive and un-tethered. This imposes substantial restrictions on the amount of hardware that can be placed on these devices. In these large sensor network systems, we need nodes to be able to locate themselves in various environments, and on different distance scales. Bulusu *et. al.* discuss idealized radio model and localization algorithm for this scenario [30]. Ward *et. al.* discuss a position calculation methodology referred to as multilateration using some sensors which give the range distance only [31].

In a group of robots the information from other robots about the location of a robot needs to be combined to find a final location. The problem of cooperative localization is the problem of fusing the information provided by different robots. Fusion of information can result in degradation of information if it is not done carefully. Some approaches use some sort of weighted average, often implemented as Kalman filter. Roumeliotis *et. al.* discuss collective localization of heterogeneous colony of robots using a distributed Kalman filter approach [32], [33], [34]. Madhavan *et. al.* discuss a distributed extended Kalman filtering algorithm for localization of a team of robots operating on outdoor terrain [7]. Howard *et. al.* describe a localization approach for mobile robot teams using maximum likelihood estimation(MLE) technique [5]. In MLE approach, they determine the set of estimates *(H)* that maximizes the probability of obtaining the set of current observations *(O)*; i.e., they seek to maximize the conditional probability $P(O|H)$. However, all these methods do not typically provide a robust solution in the presence of outliers. One way to deal with outliers and false positives is to implement some form of voting scheme like Markov Localization [35] to filter out outliers. However depending on how the Markov filter is tuned, outliers could still be allowed to affect the result, or valid observations might be discarded. Gutmann et. al. compares different localization methods using Kalman Filtering(KF), grid based Markov Localization(ML), Monte Carlo Localization(MCL) and their combinations [27].

Fuzzy logic has also been used in solving the localization problem [26], [36], [37] [38], [39]. Cooperative object localization using multiple robots using fuzzy logic to combine the location information about the object is dealt in [26]. Fuzzy logic allows combining the information provided by different robots in order to reach an agreement. In [26], two dimensional problem of locating an object by several robots in the RoboCup domain is implemented. Here, fuzzy positional information is represented in a position grid with a number associated with each cell representing the degree of possibility that the object is in the cell.

In this work, the factors which affect the pose estimation uncertainty and unreliability are identified and studied. Fuzzy sets are constructed which incorporate these uncertainties. All such fuzzy sets representing the pose estimates given by all other robots are combined using fuzzy combination rules to give a final pose estimate for each robot.

Organization of the work

In Chapter II, we formulate the problem clearly, discussing some of the issues of the problem. We also discuss the problem scenario. At the end, some of the major localization techniques, which can be used to solve the localization problem, are discussed. Chapter III deals with the Kalman filter approach and its applicability to the multi-robot localization problem. Chapter IV presents the main matter of the work. Here, fuzzy logic basics are discussed and the appropriateness of the approach towards solving the multi-robot localization is discussed. Then the main component modules of the robot are discussed. Finally, the localization procedure is described in detail. Chapter V presents the simulation in MATLAB and the results. After that, we discuss the comparison between the fuzzy logic approach and the Kalman filter approach. Chapter VI summarizes the work and concludes it.

CHAPTER II

PROBLEM

Introduction

As discussed in Chapter I, localization in a multi-robot scenario is a very important problem. Researchers have done extensive work towards localizing multiple robots in different scenarios and environments. Physical landmarks present in the environment can help in localization, but in many cases, they have to be modified or instrumented so that the robots can identify them. GPS is a very good tool for localization, but it is unavailable in many indoor environments due to signal obstruction. Global overhead camera can also be used effectively in indoor environment, but it may not be always feasible in complex indoor environments. In this work, we consider an environment where there are no landmarks and there is no access to any global positioning system (GPS) or global overhead camera. Localizing multiple robots can be done by simply locating each one of the robots individually, but there is an inherent advantage in this multi-robot scenario. Robots can cooperate with each other by sharing information to locate each other. Each robot can give pose estimates for other robots. These pose estimates from other robots can be used to compensate for the odometry errors. These pose estimates need to be combined to obtain a final pose value in such a way that it should be as close to the actual pose value of the robot.

Problem statement

Given a group of robots, each one capable of measuring

(a) changes in its own pose (position (x, y) and orientation (θ)) using odometers and

(b) the distance vector to other robots from itself using omni-directional stereo camera,

apply fuzzy logic to model the reliability of its pose estimates given by all other robots and combine these fuzzy estimates to calculate its final pose without using landmarks.

Basic robot components

The robot, as the problem statement directs, should have some basic components. So, we give a description of the basic components of the robot. The robot consists of two wheels at the front and one castor wheel at the back. Each front wheel is connected to a motor which drives it. The front wheels also have optical wheel encoders (proprioceptive sensors) attached to them as explained in Chapter I. These encoders can be used to find the number of rotations of the two wheels. The number of rotations can be used to calculate the change in the robots pose. The robot also has an omni-directional stereo camera (exteroceptive sensor) mounted on it. This camera setup is used to find the range vector of the other robots. The robot also has a transmitter and a receiver to communicate with other robots. There is a processing unit for executing the localization algorithm.

Problem scenario

A sample case of six robots is considered here in this work. The robots can translate and rotate about their body axis. Fig. 4 shows the top view of the robots.

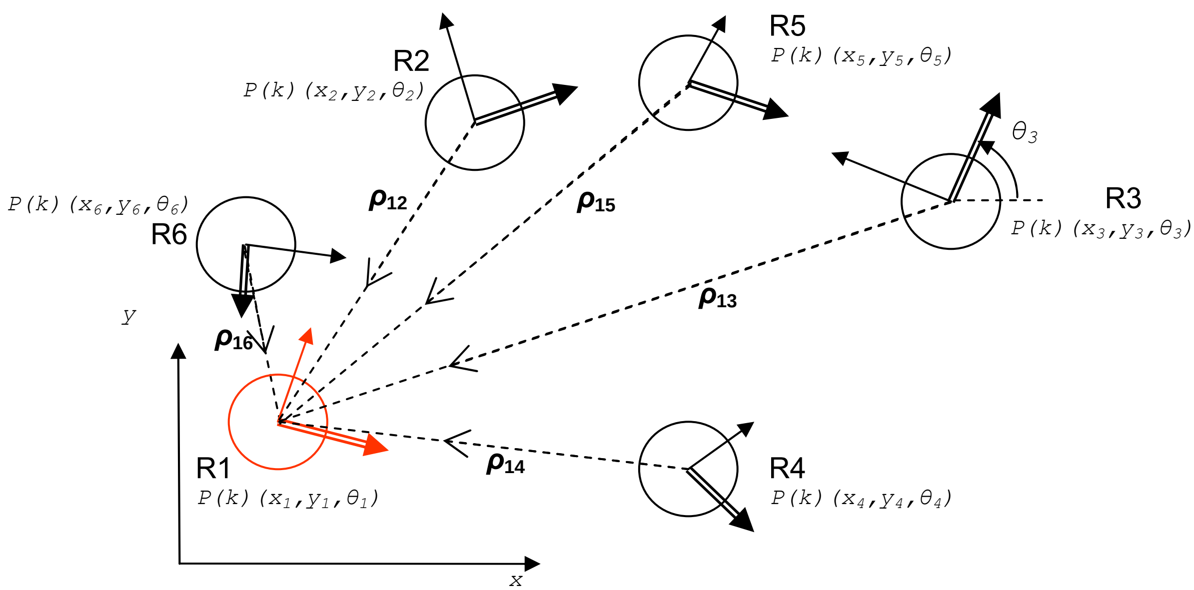


Fig. 4. The problem scenario. There are 6 robots in this example. At this instant, all the robots R2 to R6 are giving pose estimate for R1.

At this instant (see Fig. 4.), all the robots R2 to R6 give a pose estimate for robot R1.

The robots are represented as circles here. All the robots have a coordinate system attached to them, which is represented by two arrows, the double arrow being the x-axis and the other single arrow being the y-axis. There is a fixed global reference coordinate system. The wheel encoders are used to measure the angular displacements of the wheels and the omni-directional stereo camera to measure the range vector to other robots.

Here,

P(k) (x_i, y_i, θ_i) represents the pose of robot Ri at instant k.

ρ_{ij} is the range vector of robot Ri, as measured by the omni-directional camera (exteroceptive sensor), with respect to robot Rj's reference frame.

The main problem dealt here is how to combine the range vector ρ_{ij} for Ri and the pose P(k) (x_j, y_j, θ_j) by a Rj to obtain a pose estimate for Ri. And, finally how to combine all these estimates by all Rj's to find a final value of pose P(k+1) (x_i, y_i, θ_i).

Main issues of the problem

The data from the odometry sensors of a robot and the range sensors attached to all other robots contains errors. These errors need to be properly incorporated in the data representation. Also, these data have to be combined together to calculate the pose of each robot. The main issue in the problem of cooperative localization is how to fuse or combine the information provided by different robots. Fusion of this information can improve the perception of each individual robot, but, if not carefully done can result in degradation of information. For example, accurate and correct estimate for R1 given by

R2 combined (using some sort of weighted average method) with an inaccurate estimate for R1 given by R3 will always be worse than estimate of R2 by R1 alone. This problem of fusion is typically very significant in the presence of outlier robots. So, fusion of information should be done very carefully. For fusing the various pose estimates, they have to be first, represented or modeled, taking care of the uncertainty and unreliability associated with it.

Conventional localization approaches

There are various conventional approaches which deal with localization in a multi-robot scenario. Some of the basic approaches proposed in the literature are as follows:

1. *Global Positioning System (GPS):* GPS communicates with satellites to determine latitude, longitude and elevation. Every robot would have the GPS attached to it, so that it can find its current absolute location. GPS is a powerful tool for localization but is generally unavailable due to signal obstructions in many indoor environments.
2. *Using global overhead camera:* Localization can be done using a global overhead camera. Using this camera, all the robots can be seen and their actual locations can be found out. This is very suitable for a small indoor environment. But having a global camera system may not be possible always especially when the robot has to move around in large indoor complex environment.

3. *Landmark based localization:* If we know the locations of the landmarks, we can use this data to locate moving robots. Landmarks are features in the environment that a robot can detect. Sensor readings from a robot are analyzed for the existence of landmarks in it. Once landmarks are detected, they are matched with a-priori known information of the environment to determine the position of the robot. Landmarks can be divided into active and passive landmarks. Active landmarks, also known as beacons, are landmarks that actively send out location information. A robot senses the signals sent out by the landmark to determine its position. If the landmarks do not actively transmit signals, the landmarks are called passive landmarks. The robot has to actively look for these landmarks to acquire position measurements.

   This approach requires prior models of the environment which is generally unavailable, incomplete or inaccurate. Also, this requires the robots to identify and recognize the landmarks so in many cases, the landmarks have to be instrumented (artificial marks or signs are placed on the landmarks).

4. *Using portable landmarks:* The whole group of robots is divided into two groups. One group is forced to be stationary for some time and then the locations of the other group robots are used to locate the moving robots. After some time, the role is reversed. This approach limits the mobility of the group.

5. *Using maximum likelihood approach:* In this approach, we maximize the set of pose estimates *(H)* that will most likely give rise to the current observations *(O)* done by different sensors attached to the robot, i.e., we seek to maximize the conditional probability $P(O|H)$.
6. *Kalman filter approach:* It optimally combines the pose estimates given by all the other robots to calculate the pose of each robot. The pose estimates are assumed to be Gaussian. Gaussian density function is fully characterized by two parameters, the mean and the variance. The Gaussian assumption might not always be practically true, but it allows the Kalman filter to efficiently make its calculations. If the estimates are not drastically incorrect and are represented as normal distributions, Kalman filter approach produces good results.

The above mentioned approaches do not take care of any outlier robot estimate very well. An outlier robot is the one which gives a pose estimate which is drastically different from the actual estimate. This error may be due to many physical parameters but the dependency on these factors can't easily be determined accurately. The approaches mentioned above, rather combine the outlier reading to find a final estimate by some kind of weighted averaging.

The fuzzy logic approach towards solving this localization problem developed in this work is quite robust in the presence of outliers. In the next chapter, we describe a basic version of Kalman filter approach for localization. In Chapter V, we compare the performance of the Kalman filter approach and the fuzzy logic approach developed in this work.

CHAPTER III

KALMAN FILTER APPROACH

Introduction

The Kalman filter (KF) is a mathematical tool to estimate the state of a noisy dynamic system using noisy measurements related to the state. In the context of the problem discussed, the KF can be described as a technique from estimation theory that combines the information of different uncertain sources to obtain the values of variables of interest together with the uncertainty in them. The fact that the variables of the state might be noisy and not directly observable makes the estimation difficult. To estimate the state a KF has access to measurements of the system. These measurements are linearly related to the state and corrupted by noise. If these noise sources are Gaussian distributed, then the KF estimator is statistically optimal with respect to nay reasonable measure for optimality. The KF processes all available measurements to estimate the state, both accurate and inaccurate ones. KF has been successfully applied in many applications, like missions to Mars, and automated missile guidance systems. In this chapter we consider the approach and discuss the localization algorithm implemented.

Background and basics

The Kalman filter can be represented as a set of mathematical equations that provides an efficient computational means to estimate the state of a process. The discrete time Kalman filter [40], addresses the general problem of trying to estimate the state $x \in R^n$ of a discrete-time controlled process that is governed by the linear stochastic difference equation

$$x^k = A\,x^{k-1} + B\,u^{k-1} + w^{k-1}$$

with a measurement $z \in R^n$ that is

$$z^k = H\,x^k + v^k$$

The random variables w^k and v^k represent the process and measurement noise respectively. They are assumed to be independent (of each other) and with normal probability distributions,

$P(w) = N\,(0, Q)$ *zero mean and variance Q*

$P(v) = N\,(0, R)$ *zero mean and variance R*

The Kalman filter can be described as a prediction-correction approach [40] as explained below. There are two phases, first one is the prediction phase in which the states are predicted based on the state values at previous iteration. The second one is the correction phase, in which the states are corrected by incorporating the measured value of state. Note that the states are not crisp values but instead, represented as normal distributions with a mean value and some variance.

Assuming, no control input,

Prediction phase:

$$x^{k-} = A\, x^{k-1}$$

$$P^{k-} = A\, P^{k-1} A^{T} + Q$$

Correction Phase:

$$z^{k} = H\, x^{k} + v^{k}$$

$$x^{k} = x^{k-} + K\, (z^{k} - H\, x^{k-})$$

$$P^{k} = (I - K\, H)\, P$$

Where,

$$K = P^{k-} H^{T} (H\, P^{k-} H^{T} + R)^{-1}$$

Kalman filter applied to the localization problem

The Kalman filter approach can be applied to the localization problem discussed here [7], [32], [33], [41]. Negenborn describes the Kalman filter approach applied to localization [41]. A simplistic version of the Kalman filter approach is described in this chapter. Here we assume that at every instant of localization, all the robots are stationary momentarily and the robots give pose estimates for all the other robots. The accuracy of the estimates given by a robot for other robots depends upon its pose value, which is calculated based on the odometry sensors. These pose estimates are represented as Gaussian distributions as shown in Fig 5. For a robot, all such estimates given by other robots are fused together and then a final pose value is calculated. Fusion of Gaussian distributions is dealt in [42].

So, if

$x_{ki} = N(\mu_x , \sigma_x^2)$

$z_{kj} = N(\mu_z , \sigma_z^2)$

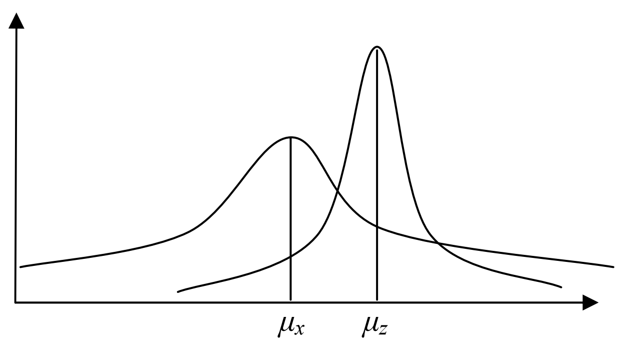


Fig. 5. Pose estimates represented as Gaussian distributions.

where,

x_{ki} is the current x-coordinate of the pose value for robot Ri at instant k,

z_{kj} is the x-coordinate of the pose estimate given by one of the robot Rj for robot Ri.

x_{ki} and z_{ki} are Gaussian distribution with μ_x and μ_z as mean values and σ_x^2 and σ_z^2 as respective variances.

then,

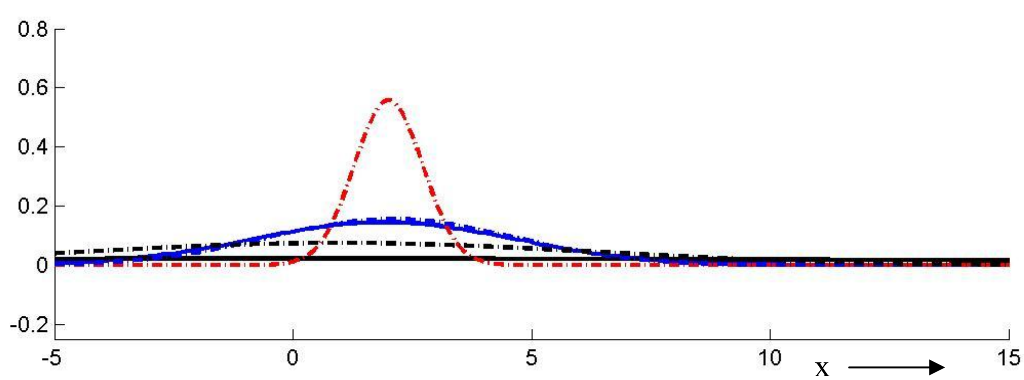
$$x_{kfi} = x_{ki} + K\,(z_{kj} - x_{ki})$$

$$\sigma_f^2 = (1 - K)\,\sigma_x^2$$

where, K is the Kalman gain given by,

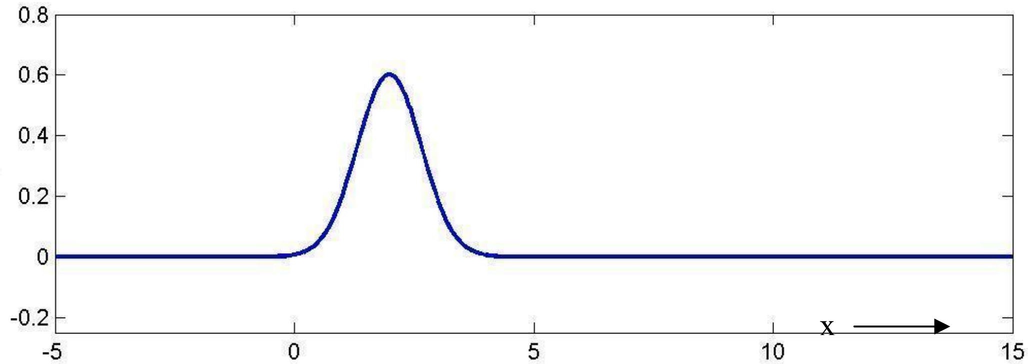
$$K = \sigma_x^2\,(\sigma_x^2 + \sigma_z^2)^{-1}$$

The above equation can be used to recursively combine the measurements (z_{kj}) provided by all the robots (Rj's) and thus obtain an optimal final value for robot Ri. This procedure is repeated for all the other pose parameters like y-coordinate and θ-coordinate values.



(a) Pose estimates given by all the other robots for one robot.

Fig. 6. Pose estimates in Kalman filter approach



(b) The pose estimates are fused together using Kalman filter approach.

Fig. 6. continued.

A sample case considered in this work demonstrates this approach very well. Fig. 6 shows the pose estimates for a robot by other five robots in a six robot example.

We can incorporate the motion model to see if it improves the accuracy of the pose calculation. The motion model, under certain assumptions mentioned in the next section makes it clear that it doesn't really improve the accuracy of the pose calculations.

Incorporating robot motion model

We can take the robot motion model into account. There is an assumption made here that every time localization is done, the variance is assumed to be zero after a final pose value is calculated. The robot motion model really doesn't affect the results by

Kalman filter under this assumption. Each robot calculates its pose estimate by integrating the velocity and acceleration as shown below.

$$x(k+1) = x(k) + v(k)*T$$

$$v(k+1) = v(k) + \alpha(k)*T$$

$$\alpha_l(k+1) = \alpha * r + noise$$

where,

x is the x coordinate of the robot

v is the x-component of the velocity of the robot

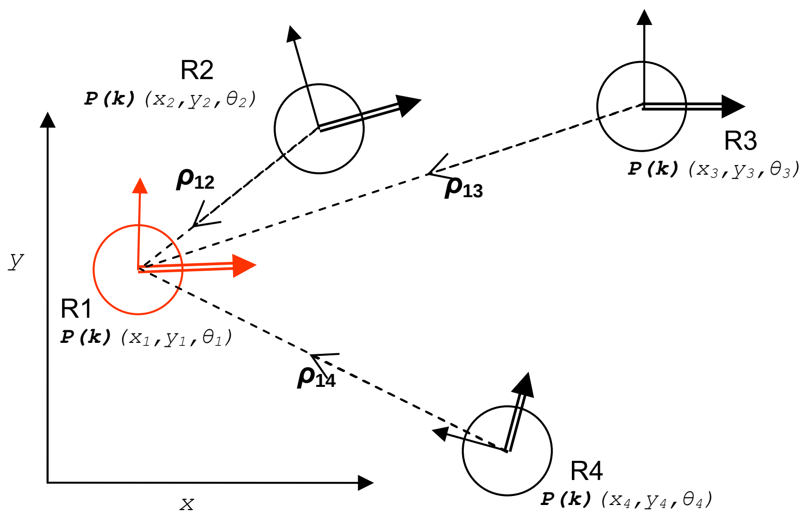
α_l is the x-component of the linear acceleration of the robot

α is the mean angular acceleration of the two wheels of the robot and

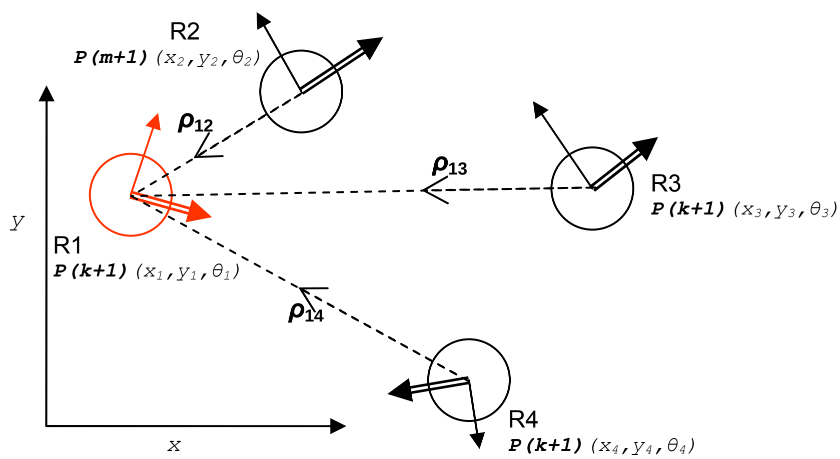
r is the radius of the wheels

The noise in the linear acceleration comes because of the odometry errors.

Let's consider the problem scenario as shown in Fig. 7:



(a) Configuration at instant k.



(b) Configuration at instant k+1.

Fig. 7. Configurations of the robot scenario.

Here, all the robots from R2 to R6 are estimating pose values for robot R1. All the robots R1 to R6 have moved from locations at kth instant to different locations at instant k+1. At instant k+1, the pose values of all the robots have some mean values and variances associated with them. Now, when the robots R2 to R6 give pose estimates for robot R1.

Now, here if the state is taken as:

$$\begin{bmatrix} x \\ v \\ \alpha_l \end{bmatrix}$$

Then,

$$\begin{bmatrix} x(k+1) \\ v(k+1) \\ \alpha_l(k+1) \end{bmatrix} = \begin{bmatrix} 1 & t & 0 \\ 0 & 1 & t \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x(k) & v(k) & \alpha_l(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \alpha * r \end{bmatrix} + \begin{bmatrix} \text{noise} \\ \text{noise} \\ \text{noise} \end{bmatrix}$$

i.e., $\mathrm{X(k+1) = A\ X(k)\ +\ B + Noise}$

Now, P(k+1) is given by:

$\mathrm{P(k+1) = A\ P(k)\ A^T + Q}$

Where, $P(k)$ is the variance associated with $x(k)$ and Q is the noise which depends upon the wheel slippage.

Now taking,

$P(k) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ since the variance is assumed to be zero after every step of localization and updating of the pose values for all the robots.

Therefore,

$P(k+1) = Q$, the uncertainty which depends upon the wheel slippage.

Therefore the motion model does play a role in getting the value of x at instant k+1 but it doesn't affect the variance associated with x. The formulation of the simple Kalman filter is useful in comparing with the fuzzy logic approach.

The Kalman filter is implemented in MATLAB and the results are compared in Chapter V with the fuzzy logic approach developed here in this work. In the next chapter, the fuzzy logic approach towards solving this localization problem is discussed.

CHAPTER IV

FUZZY LOGIC APPROACH

Introduction

The concept of fuzzy set and fuzzy logic were introduced by Zadeh [43]. Ordinarily, a set is defined by its members. An object may be either a member or a non-member: the characteristic of traditional (crisp) set. The connected logical proposition may also be true or false. This concept of crisp set may be extended to fuzzy set with the introduction of the idea of partial membership. Any object may be a member of a set 'to some degree'; and a logical proposition may hold true 'to some degree'.

Fuzzy set theory offers a precise mathematical form to describe such fuzzy terms in the form of fuzzy sets of a linguistic variable. To represent the shades of meaning of such linguistic terms, the concept of grades of membership or the concept of possibility values of membership has been introduced. We write $\mu(x)$ to represent the membership of some object in the set X. Membership of an object will vary from full membership to non-membership.

Any fuzzy term may be described by a continuous mathematical function or discretely by a set of numerical values. Having obtained the numerical representation of these linguistic terms, one has to define the set theoretic operations of union, intersection and complementation along with their logical counterparts of conjunction, disjunction and complementation as follows:

- Union (logical OR): the membership of an element in the union of two fuzzy sets is the larger of the memberships in these sets.

  (A OR B) = max ((A), (B))

  e.g., (tall OR small) = max((tall), (small))

- Intersection (logical AND): the membership of an element in the intersection of two fuzzy sets is the smaller of the memberships in these sets.

  (A AND B) = min ((A), (B))

  e.g., (tall AND small) = min((tall), (small))

- Complement (logical NOT): the degree of truth of the membership to the complement of the set is defined as (1 - membership).

  (NOT A) = 1 - (A)

  e.g., (NOT tall) = (1 - (tall))

Fuzzy logic approach to solve the problem

A stationary robot is free from odometery errors and therefore can provide the best estimate for another robot [29]. If the estimator robot is accelerating and moving fast and taking frequent turns, the odometer errors are expected to pile up and therefore, the estimate given by it is not so reliable. In a group of many robots, the robots which are moving with less velocity and less acceleration and taking fewer turns are expected to provide more reliable and accurate estimates than other ones. This reliability is modeled and is used to convert crisp pose estimates provided by other robots into fuzzy pose estimates and then combined together using fuzzy logic.

We see each robot as an expert which provides pose estimation with varying degree of reliability about other robots. This reliability being a function of the following various physical quantities:

1. *Angular acceleration of the wheels of the robot (α):* If the wheel angular acceleration is large, the wheels are more likely to slip as explained in the next section.
2. *Distance between the two robots (d):* The larger the distance of separation between the two robots the more unreliable is the pose estimate from one to another. This is due to the resolution of the omni-directional stereo camera as explained in the next section.
3. *Distance traveled by the robot since the last localization:* The larger the distance traveled by the robot since last localization, the more unreliable is the pose estimate. This is because of the fact that the uncertainty and errors keep on piling up.
4. *Number of turns taken by the robot:* When the robot takes turns especially at high speeds, it is more likely to slip.

In this work, for simplicity, we consider only the first two factors. We model the reliability of the pose estimate by converting the crisp pose value to a trapezoidal fuzzy membership function. We combine all such fuzzy membership functions using fuzzy logic techniques.

Modeling the reliability of information

The reliability of pose estimates depend upon the physical factors mentioned above. Here, we discuss in detail about the dependability of reliability of pose estimates upon the two factors namely the mean angular acceleration of the wheel of the robot and the distance of separation between the two robots.

Reliability of pose estimate and angular acceleration (α)

The velocity $\boldsymbol{v}$ and thus the displacement can be calculated by measuring $\boldsymbol{\omega}$ and using (1), provided the wheel doesn't slip on the ground as shown in Fig. 8.

$$v = \omega R \tag{1}$$

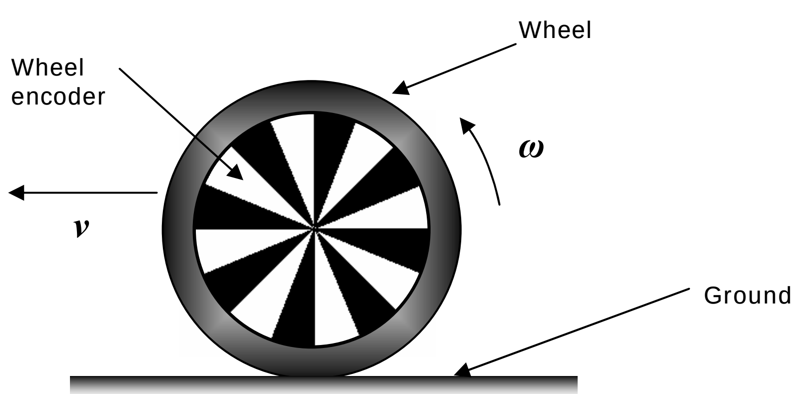


Fig. 8. Optical encoder attached to the robot wheel.

When the angular acceleration is high, the probability of wheel slippage increases. This wheel slippage makes the robot's linear displacement, which is calculated using (1), unreliable. This decreases the reliability of pose estimates for other robots by this robot.

Reliability of pose estimate and distance from the current robot (d)

The resolution of the stereo imaging camera setup decreases as the distance of the object increases [44].

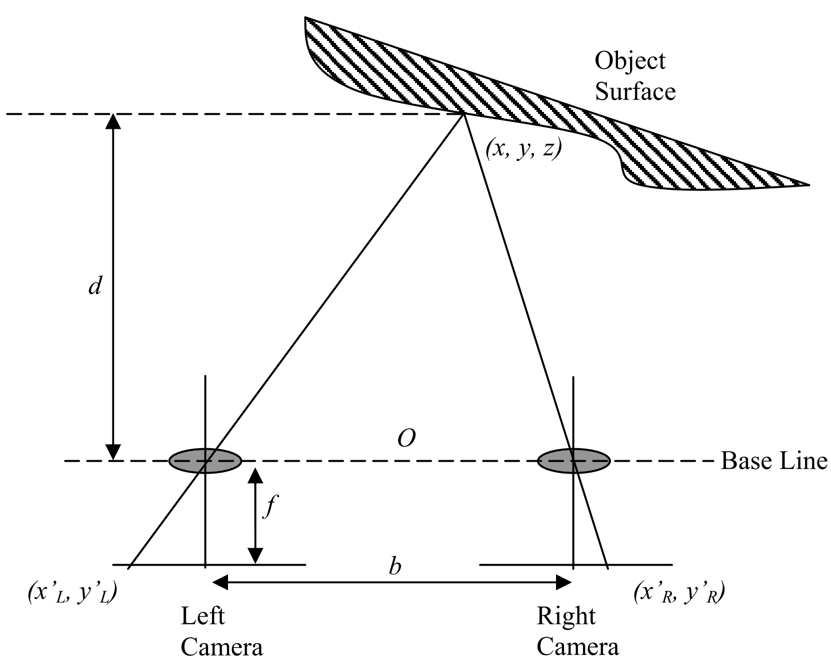


Fig. 9. Stereo camera range measurement system.

Thus, the reliability of the range vector calculated using the images from the cameras of this sensor decreases as the distance between the estimator robot and the current robot increases as shown in Fig. 9.

Component modules of the fuzzy logic approach

The detailed robot components schematic is shown in Fig. 10. These modules have different roles which are mentioned below. The odometry sensors are used to sense the pose value of a robot which is taken as the first basic crude pose estimate. Then range vector measurements are taken for all other robots. These range vectors are combined with the basic odometry based pose estimates, and pose estimates are given by each robot for all other robots. These estimates are crisp but inaccurate. So they are converted into fuzzy membership functions. The estimates given by all the robots are then finally combined to find a crisp and more accurate pose value for each robot.

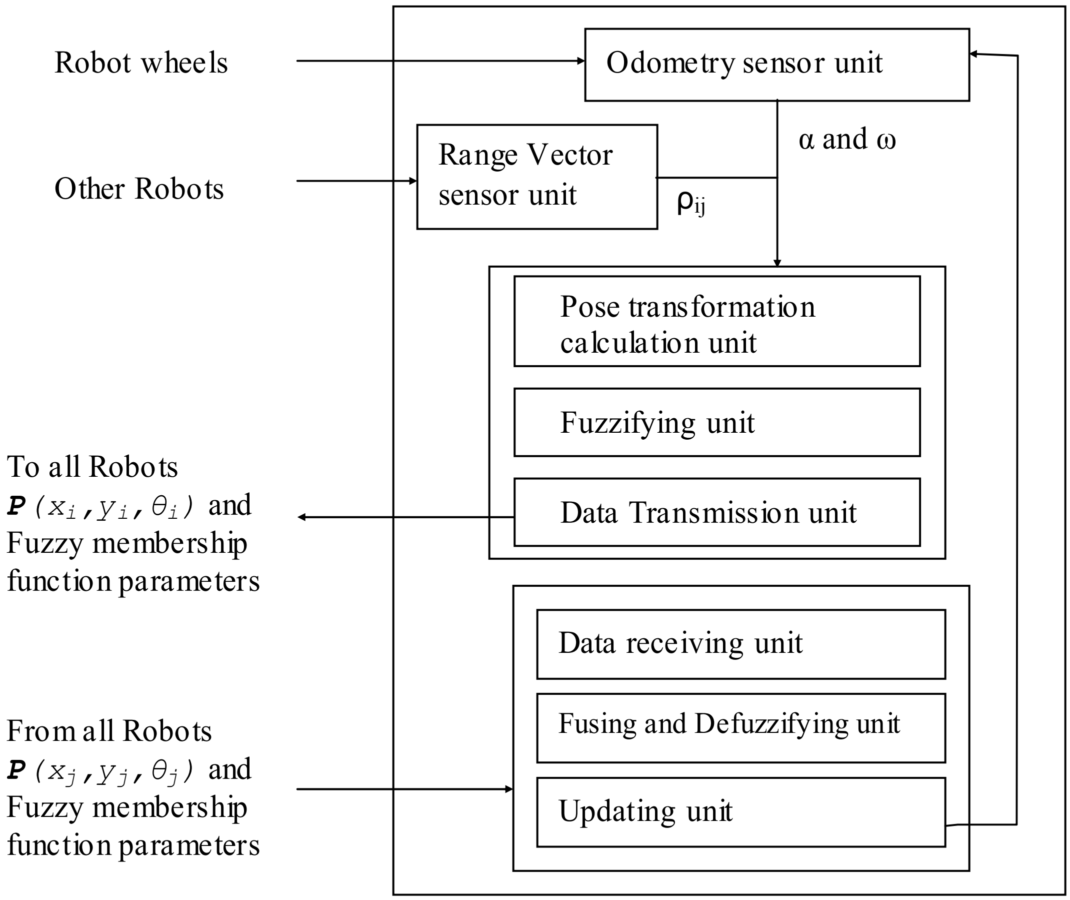


Fig. 10. Detailed robot components and procedure schematic.

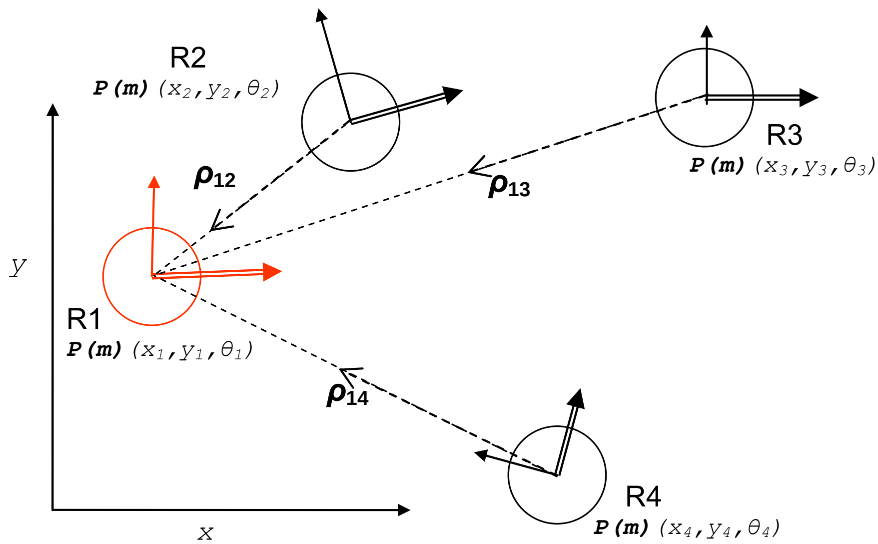
The various components mentioned in Fig. 10 are:

1. *Odometry sensor unit:* senses the robot's linear distance moved from the last iteration by integrating the wheel encoder readings.
2. *Range vector sensor unit:* perceives the range vector of the other robot.

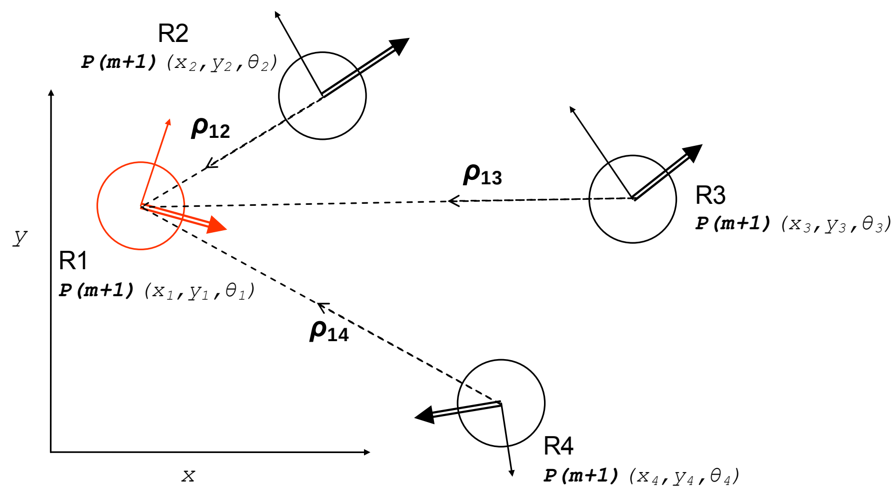
3. *Pose transformation calculation unit:* transforms the local pose estimates to the global pose estimate, that is, to the pose values with respect to global coordinate system.
4. *Fuzzifying unit:* converts the crisp values of pose estimates to fuzzy membership functions based on the output of odometry sensor unit and the range vector sensor unit.
5. *Data transmission unit:* transmits the pose values and the fuzzy membership function parameters a, b and c for each pose parameters x, y and θ.
6. *Data receiving unit:* receives the data transmitted by all the robots.
7. *Fusing and defuzzifying unit:* combines the fuzzy pose estimates given by other robots and its own pose estimate based on odometeric correction to calculate the possibility distribution for the pose and then defuzzifies to calculate a crisp pose estimate.
8. *Updating unit:* updates the pose value by the above final crisp estimate.

Procedure

The various components, as shown in Fig. 10, play different roles towards the localization process. The pose values taken here are with respect to the global axes. For figure clarity only four robots are shown in Fig. 11, which shows the problem scenario, but there are six robots in the simulation. A procedure is presented in sequential manner by describing the roles of the component modules.



(a) at instant k = m



(b) at instant k = m+1

Fig. 11. Problem scenarios at two instances.

Here, just note that, R2 is nearest to R1 and has moved very less, whereas R3 has moved a large distance and is far away from R1, so pose estimate of R1 by R2 would be more reliable than that by R3.

Odometry sensor unit

This component is used to sense the robot's angular velocity and angular acceleration and thus the distance moved from the last iteration. For wheeled robots, generally, the linear displacements and the linear velocities are calculated using the rotation of the wheels. Using the optical encoders on both the wheels to measure their angular displacements, the displacement, velocity and the acceleration of the robot can be calculated.

The calculation of linear displacement and velocity of the robot is correct if the wheels do not slip.

Range vector sensor unit

Using this, the robots determine the range vectors ($\boldsymbol{\rho_{ij}}$) of the other robots. One of the sensors which provide this data is omni-directional stereo camera [8]. This camera setup takes two images (as shown in Fig. 3), one by each camera, which is complete 360^{o} view around the robot. So, all the robots which are visible by this robot would be present in these two images. Comparing the position shifts in these two images, the actual range distances to the robots can be found out.

Pose transformation unit

The range data is with respect to the estimator robot's coordinate system, so it needs to be transformed to the global coordinate system.

The coordinates of Ri as seen from Rj are:

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} x_j \\ y_j \end{bmatrix} + \begin{bmatrix} \cos\theta_j & -\sin\theta_j \\ \sin\theta_j & \cos\theta_j \end{bmatrix} \begin{bmatrix} x_{ij} \\ y_{ij} \end{bmatrix}$$

$$\begin{bmatrix} x_j \\ y_j \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \end{bmatrix} + \begin{bmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{bmatrix} \begin{bmatrix} x_{ji} \\ y_{ji} \end{bmatrix}$$

$$\sin\theta_i = \frac{[(y_j - y_i)x_{ji} - (x_j - x_i)y_{ji}]}{R_{ij}^2}$$

$$\cos\theta_i = \frac{[(x_j - x_i)x_{ji} - (y_j - y_i)y_{ji}]}{R_{ij}^2}$$

Where,

x_i, y_i, θ_i are the global pose values for robot Ri

x_j, y_j, θ_j are the global pose values for robot Rj

x_{ij}, y_{ij} is the range vector's (R_{ij}) x and y components of Ri from Rj with respect to Rj reference coordinate system

x_{ji}, y_{ji} is the range vector's (R_{ji}) x and y components of Rj from Ri with respect to Ri reference coordinate system

Fuzzifying unit

The final global estimated values of the pose parameters depend upon the acceleration of the estimating robot and the distance of separation of the estimating and the current robot. This dependency is represented as a trapezoidal fuzzy membership function as shown in Fig. 12. The lower the value of a, the lower would be the value of b and c. Low values of a, b and c represent a reliable crisp value of x1. Large values of a, b and c means that the value of x1 is more unreliable.

Assuming,

$$b = a/k1 \qquad \text{and} \qquad c = a*k2 \tag{2}$$

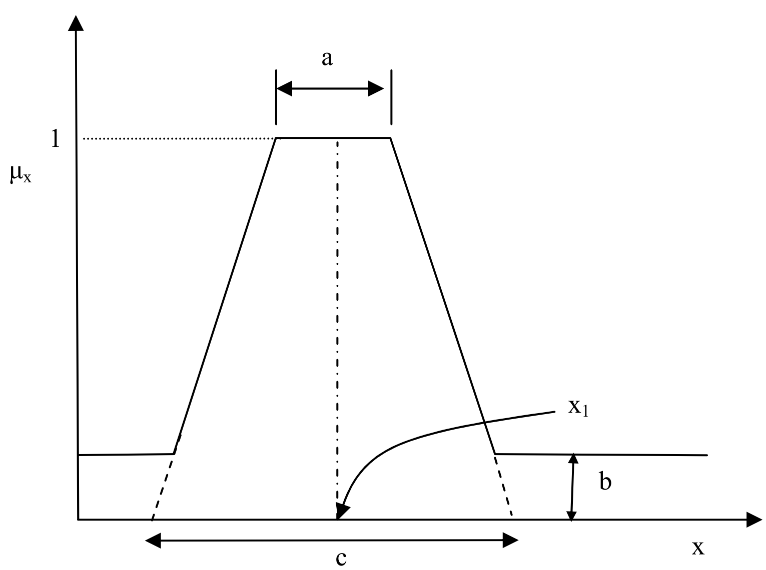


Fig. 12. Trapezoidal fuzzy membership function. Converting a crisp value x1 to a trapezoidal fuzzy membership function

Determination of the values of trapezoindal fuzzy set characteristic parameters

The assumption of the dependency stated at (2), makes it sufficient to determine the value of a, which can be calculated using fuzzy rules. Fig. 13 shows the fuzzy rules in a matrix form.

a		**α**		
		Small	Medium	Large
d	Small	Very Small	Small	Large
	Medium	Small	Medium	Large
	Large	Large	Large	Very Large

Fig. 13. Fuzzy rule matrix. **α** is the mean angular acceleration of the wheels of the robot and **d** is the distance of separation between two robots

Fusing and defuzzifying unit

The various fuzzy pose estimates are then combined (fused together), using fuzzy membership combination techniques, to calculate the possibility distribution (p. d.) of the pose of the robot.

The fuzzy pose estimates are combined using two operators:

1. min as a combination operator
2. product as a combination operator

Finally, crisp values are calculated by defuzzifying the p. d. by the following approaches as shown in Fig 14.

1. MOM: mean of maximum
2. COA: center of area
3. Cutoff COA: applying a threshold to the p.d. and then applying COA.

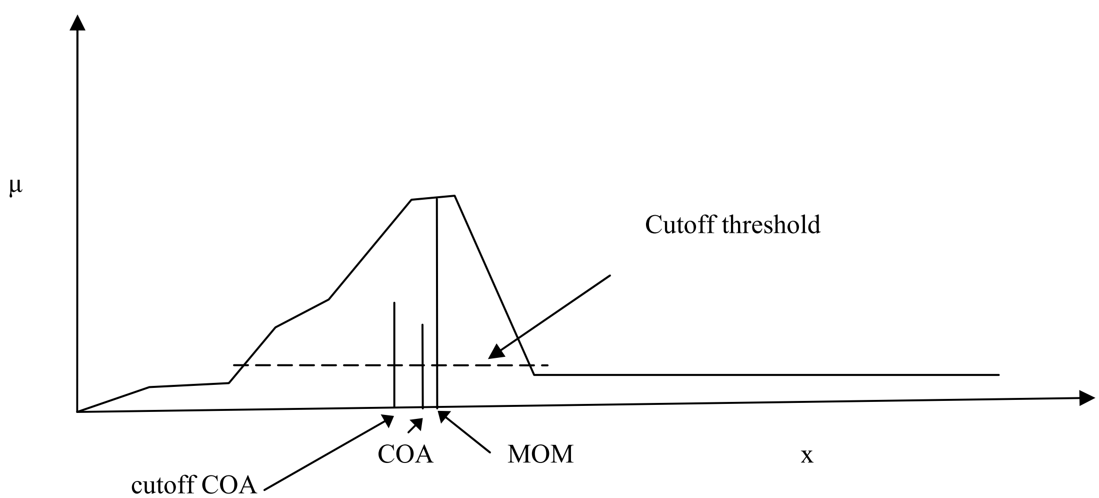


Fig. 14. Defuzzification of the possibility distribution function. Various approaches like COA, MOM and cutoff COA is shown here.

The fuzzy logic approach towards localization problem has many advantages [45]. There are two main features of fuzzy logic that are interesting in this application (i) the flexibility of fuzzy sets to represent different types of uncertain information and (ii) the availability of different combination operators to perform the data fusion step.

The fuzzy logic approach provides a more robust solution to multi-robot localization problem in the presence of outliers. This would become clearer in Chapter V.

CHAPTER V

SIMULATIONS, RESULTS AND COMPARISONS

Simulation

An example of six robots, as shown in Fig. 15, is taken and their pose values are defined initially. Coding of the algorithm and simulation is done in MATLAB. A sample work space of 10 by 10 units is taken for simulation.

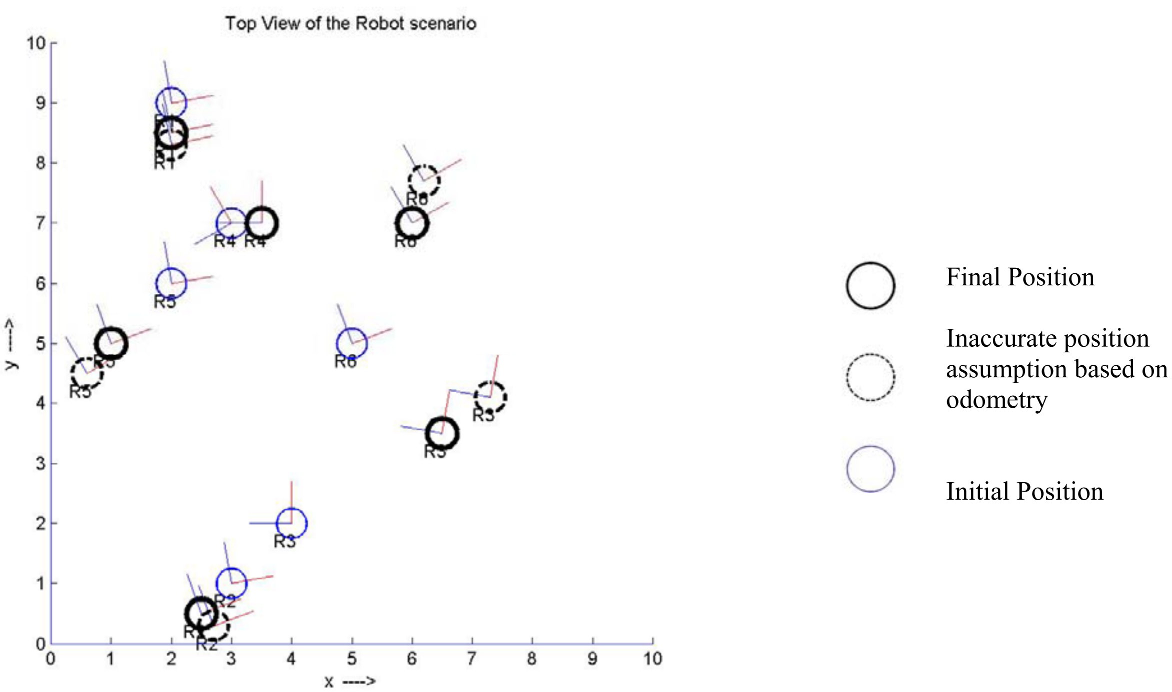


Fig. 15: Top view of the robot configuration.

Based on the various factors discussed in Chapter IV, the fuzzy pose values are calculated and then each robot predicts pose estimate for every other robot. These estimates are combined together to obtain a final crisp value for each robot.

As shown in Fig. 16, all the robots R2, R3, R4, R5 and R6 give pose estimates for robot R1. These fuzzy pose estimates are constructed on the basis of the acceleration of the estimator robot and the distance of separation between the two robots. Here, the pose estimate give by robot R4 is the most accurate which is evident by the smallest width of the peak of the function (value of 'a' is least in this case). On the contrary, the pose estimate give by robot R3 is the least accurate which is evident by the largest width of the peak of the function (value of 'a' is largest in this case). All other estimates fall in between these two extreme cases.

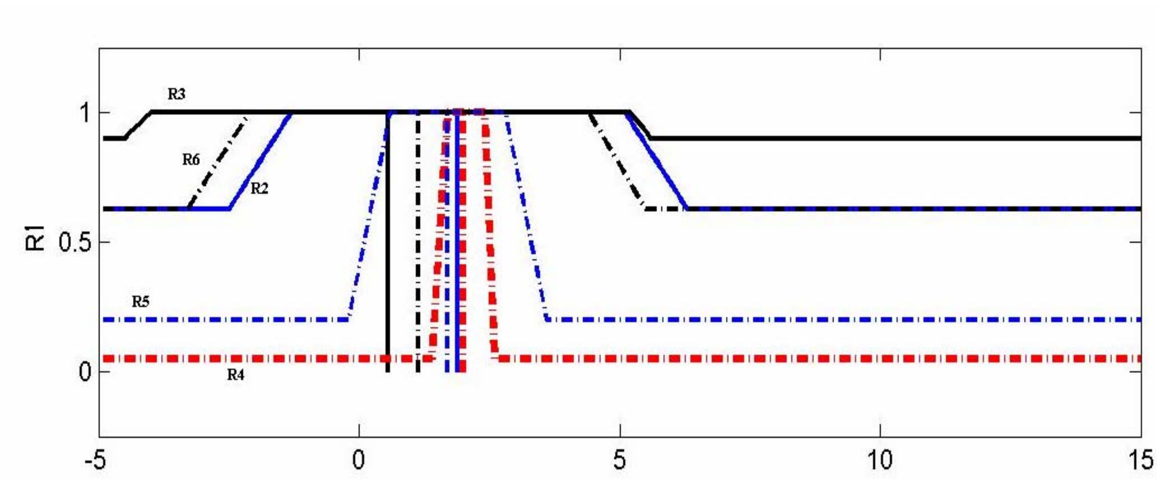
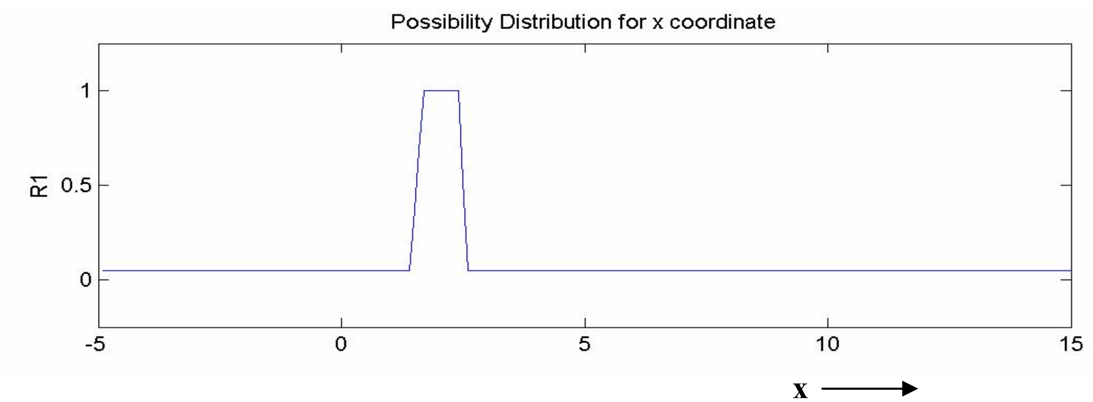
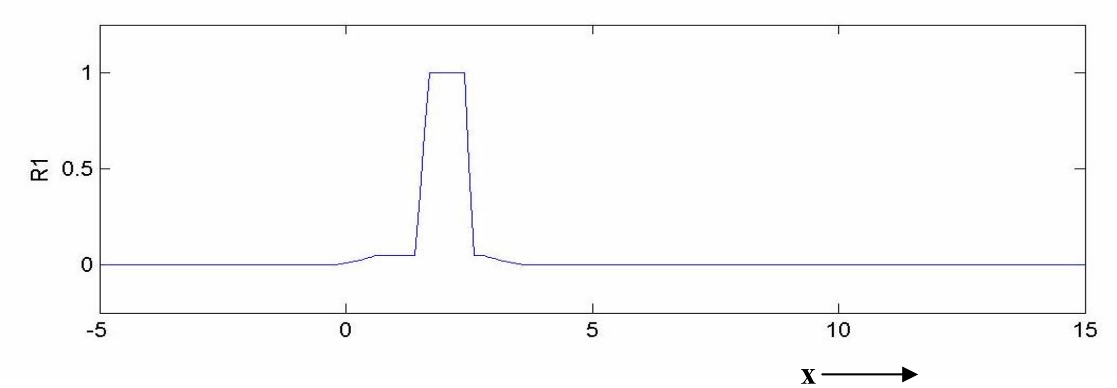


Fig. 16. The fuzzy membership functions for pose estimates. The fuzzy membership functions for x-coordinate estimate by robots R2, R3, R4, R5 and R6 for R1.

The pose estimates are fused together using fuzzy logic using ‘min’ and ‘product’ as combination operators. The results are shown in Fig. 17. The results which are seen in Fig 17 (a) and (b) are obtained by taking the minimum and product respectively of all the values at every value of x.



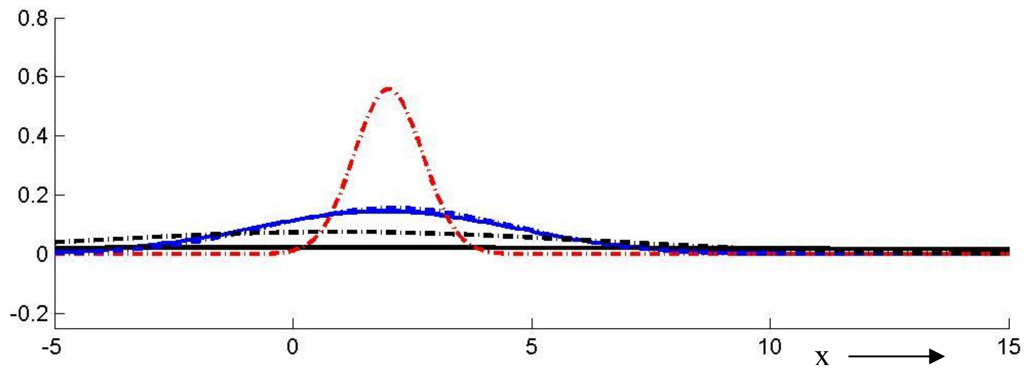
(a)



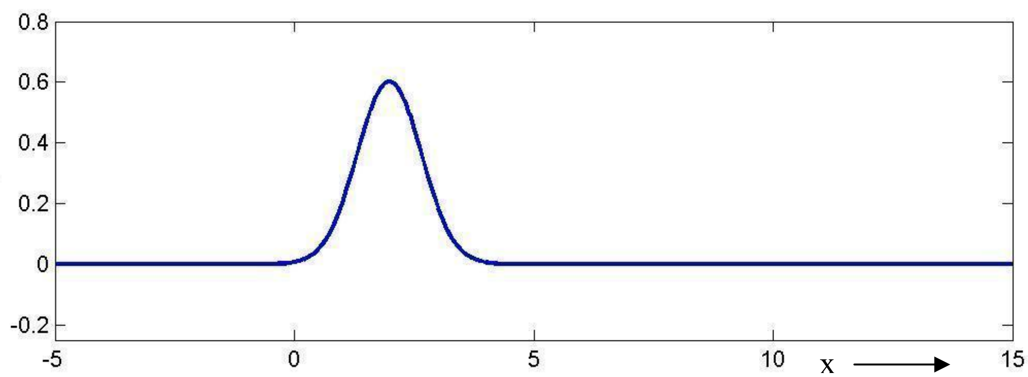
(b)

Fig. 17. Fusion of fuzzy estimates. The estimates are fused together using (a) min and (b) product approach

In Kalman filter approach also, the pose estimates are fused together as shown in Fig. 18. The fusion here is based on the method discussed in Chapter III.



(a)



(b)

Fig. 18. (a) Pose estimates given by all the other robots for R1. (b) The pose estimates are fused together using Kalman filter approach.

The fuzzy logic approach and the Kalman filter approach both are implemented and the robot scenarios are simulated on MATLAB.

As can be seen in Fig. 18, the resultant distributions of the pose estimate fusion is always a Gaussian distribution in the case of Kalman filter approach. But in the fuzzy logic approach, the resultant distribution need not be in the trapezoidal form.

Results and comparisons

The simulation results are analyzed in this section. Both the fuzzy logic and the Kalman filter approaches are compared qualitatively as well as quantitatively. Qualitative comparison demonstrates the limitations and advantages of the two approaches based on the general outline of the methodology of the approaches. Quantitative comparison is done with the simulation data. The most commonly used Kalman's filter techniques make the assumption of having a linear model of the robot and of the sensors. The fuzzy logic technique does not need these assumptions. An Extended Kalman Filter (EKF) technique, enriched by specialized routines to deal with the case of total uncertainty, can give good solutions to the problem. However, this solution would be much more complex than the fuzzy logic one [45].

Qualitative comparison

The following qualitative differences are there in the two approaches, because of the inherent differences in modeling and fusion of the pose estimates:

1. There is a consensus between the different sources of information in fuzzy logic approach rather than a tradeoff in standard probabilistic combination techniques (as can be seen from Fig 19) [26]. The final possibility

distribution very much agrees with both the distributions μ1 and μ2. In Fig 19 (b), the final distribution is not fully agreed upon by the two distributions μ1 and μ2, rather it is their weighted average.

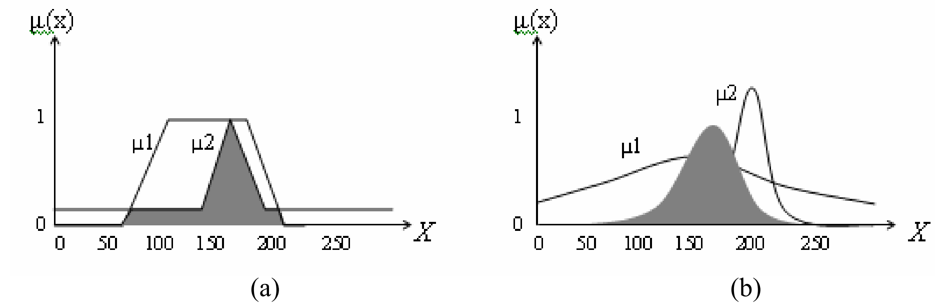


Fig. 19. Consensus versus tradeoff. (a) fuzzy logic approach of fusion (b) Kalman filter approach of fusion.

2. The fuzzy logic approach of fusion discounts unreliable information as seen in Fig 20. Information μ1 is unreliable, as indicated by the high bias, and therefore has only a small impact on the result [26].

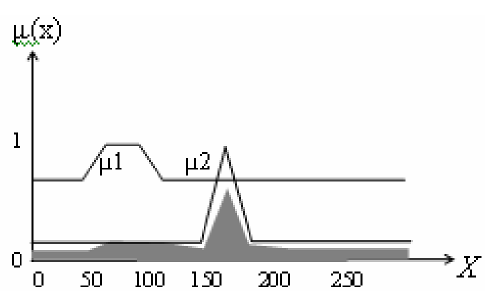
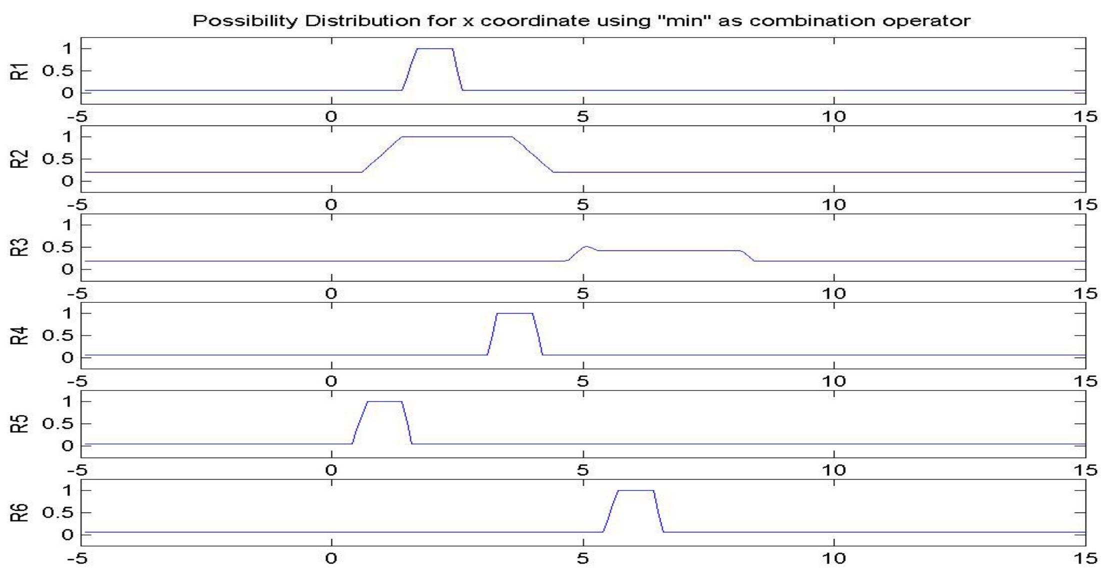


Fig. 20. Discounting unreliable information.

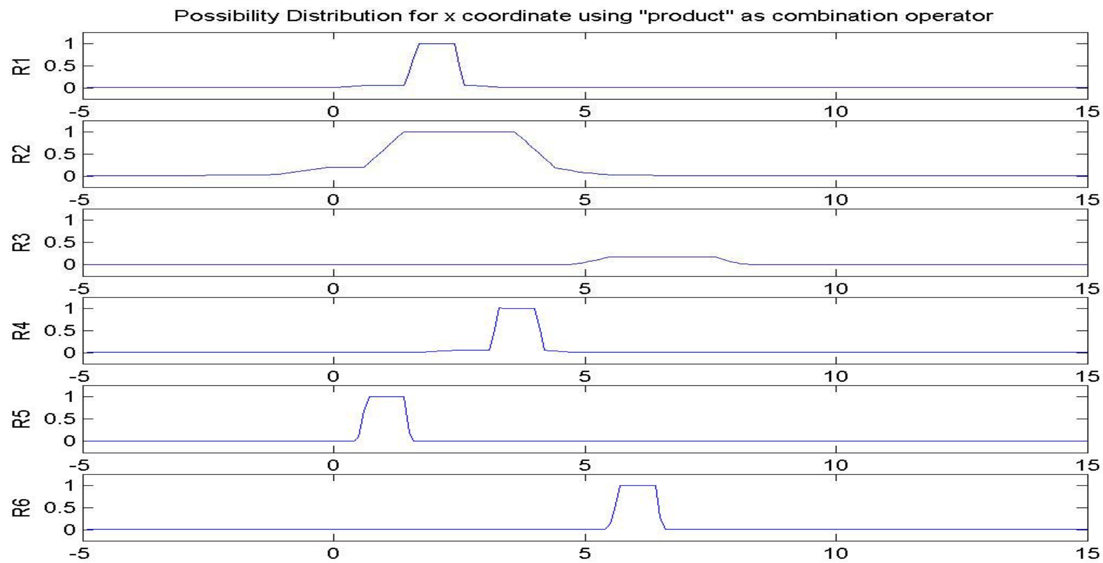
Quantitative comparison

The fuzzy logic approach is compared with the Kalman filter approach quantitatively. Fig. 21 (a), (b) and (c) show the fused x-coordinate estimates for all the six robots graphically using the min operator fuzzy logic approach, product operator fuzzy logic approach and Kalman filter approach respectively.

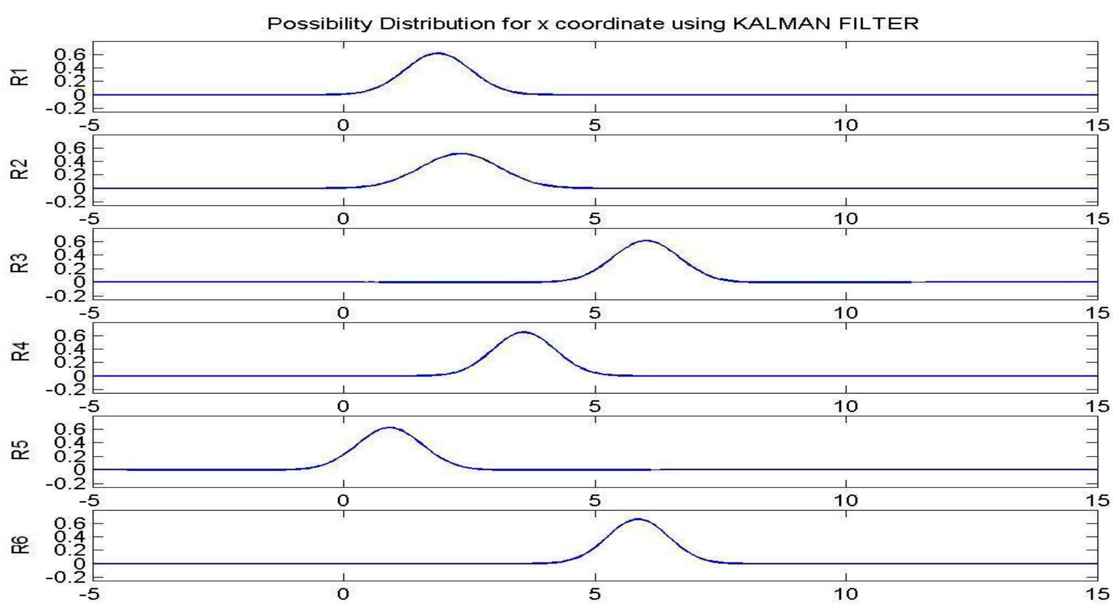


(a)

Fig. 21. Graphical representation of the possibility distribution function.



(b)



(c)

Fig. 21. continued

Pose estimate comparison

The x-coordinates estimated by all the robots are fused together and represented as possibility distributions as shown in Fig. 21. These possibility distributions are then defuzzified to find the final crisp values which are shown in TABLE 1 (a), (b), (c), (d), (e) and (f) for various accelerations of robot R3. As the acceleration of robot R3 increases, it is expected to provide more inaccurate pose estimates for all the other robots.

TABLE 1. Pose estimation comparison

Robot No.	Actual values	Fuzzy Approach (using cutoff COA)	Kalman Filter Approach	Error in F.L estimate	Error in K.F estimate
1	2.0	1.9492	1.7851	-0.0508	-0.2149
2	2.5	2.5016	2.7776	0.0016	0.2776
3	6.5	6.5685	6.6761	0.0685	0.1761
4	3.5	3.6037	3.4883	0.1037	-0.0117
5	1.0	0.9606	1.0079	-0.0394	0.0079
6	6.0	6.0402	5.9352	0.0402	0.0648
Root Mean Square error				0.1456	0.3983

(a) R3’s acceleration =1

TABLE 1 [continued]

Robot No.	Actual values	Fuzzy Approach (using cutoff COA)	Kalman Filter Approach	Error in F.L estimate	Error in K.F estimate
1	2.0	1.9492	1.8764	-0.0508	-0.1236
2	2.5	2.5016	2.9157	0.0016	0.4157
3	6.5	6.5685	6.6761	0.0685	0.1761
4	3.5	3.6037	3.5969	0.1037	0.0969
5	1.0	0.9606	1.0876	-0.0394	0.0876
6	6.0	6.0402	6.0492	0.0402	0.0492
Root Mean Square error				0.1456	0.4884

(b) R3's acceleration = 2

Robot No.	Actual values	Fuzzy Approach (using cutoff COA)	Kalman Filter Approach	Error in F.L estimate	Error in K.F estimate
1	2.0	1.9492	1.8936	-0.0508	-0.2149
2	2.5	2.5016	3.0136	0.0016	0.2776
3	6.5	6.5685	6.6761	0.0685	0.1761
4	3.5	3.6037	3.6086	0.1037	-0.0117
5	1.0	0.9606	1.1384	-0.0394	0.0079
6	6.0	6.0402	6.0492	0.0402	0.0648
Root Mean Square error				0.1456	0.5826

(c) R3's acceleration =3

TABLE 1 [continued]

Robot No.	Actual values	Fuzzy Approach (using cutoff COA)	Kalman Filter Approach	Error in F.L estimate	Error in K.F estimate
1	2.0	1.9492	1.8198	-0.0508	-0.1802
2	2.5	2.5016	2.8418	0.0016	0.3418
3	6.5	6.5685	6.6761	0.0685	0.1761
4	3.5	3.6037	3.5177	0.1037	0.0177
5	1.0	0.9606	1.0479	-0.0394	0.0479
6	6.0	6.0402	5.9677	0.0402	-0.0323
Root Mean Square error				0.1456	0.4289

(d) R3's acceleration =4

Robot No.	Actual values	Fuzzy Approach (using cutoff COA)	Kalman Filter Approach	Error in F.L estimate	Error in K.F estimate
1	2.0	1.9492	1.8185	-0.0508	-0.1815
2	2.5	2.4950	2.8271	-0.0050	0.3271
3	6.5	6.5685	6.6761	0.0685	0.1761
4	3.5	3.6037	3.5177	0.1037	0.0177
5	1.0	0.9606	1.0469	-0.0394	0.0469
6	6.0	6.0402	5.9716	0.0402	-0.0284
Root Mean Square error				0.1457	0.4174

(e) R3's acceleration =5

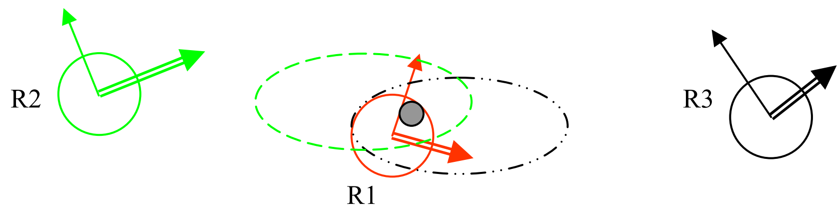
TABLE 1 [continued]

Robot No.	Actual values	Fuzzy Approach (using cutoff COA)	Kalman Filter Approach	Error in F.L estimate	Error in K.F estimate
1	2.0	1.9492	1.8229	-0.0508	-0.1771
2	2.5	2.5016	2.8354	-0.0016	0.3354
3	6.5	6.5685	6.6761	0.0685	0.1761
4	3.5	3.6037	3.5237	0.1037	0.0237
5	1.0	0.9606	1.0527	-0.0394	0.0527
6	6.0	6.0402	5.9815	0.0402	-0.0185
Root Mean Square error				0.1456	0.42.26

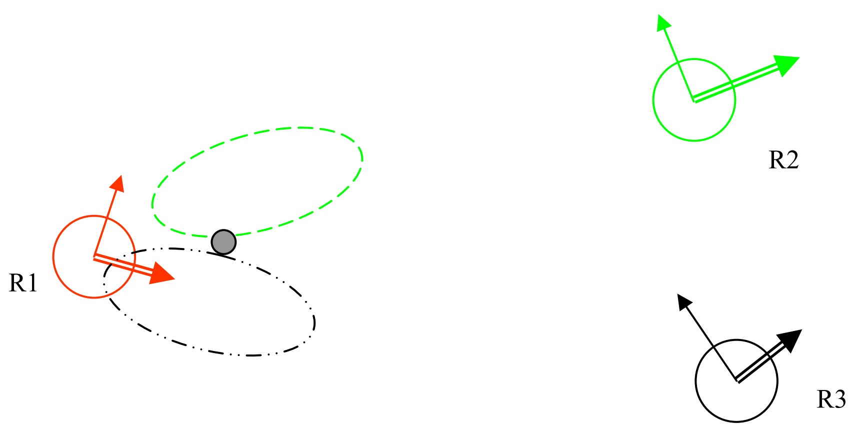
(f) R3's acceleration =6

Result interpretations and discussions

If the data in TABLE 1 is closely observed, we see that the RMS error is always smaller in case of fuzzy logic approach. In most cases individually also, the error is less in case of fuzzy logic approach than in Kalman filter approach. But, in some cases the Kalman filter approach gives more accurate pose estimates for some robots. The reason for this anomaly is basically that the Kalman filter calculates weighted average estimate value as shown in Fig. 22.



(a)



(b)

Fig. 22. Accuracy of Kalman filter approach. The ellipses represent the uncertainties associated with the pose value. The small solid grey circle represents the final pose value for the robot R1

So in case there are two robots R2 and R3 as shown in Fig. 22, giving pose estimate for robot R1, the spatial configuration arrangement contributes significantly towards the accuracy of the approach. In case (a), the pose estimate is nearer to the actual value rather than in (b).

The Kalman filter and the fuzzy logic approach results are compared in tabular form in TABLE 2.

TABLE 2. RMS error tabular comparison.

Acceleration of R3	RMS F.L error	RMS K.F error
1	0.1456	0.3983
2	0.1456	0.4884
3	0.1456	0.5826
4	0.1456	0.4289
5	0.1457	0.4174
6	0.1456	0.4226

A graphical representation of the results is shown in Fig. 23. The fuzzy logic approach performs better than a Kalman filter in the presence of the outlier robot R3. RMS error in the Kalman filter approach increases as the acceleration of R3 increases but later on flattens as can be seen in Fig 22. It is due to the fact that when R3's acceleration is not that large, the Kalman gain is affected by this outlier robot (R3) and thus the final estimate is also affected by it. But as R3's acceleration becomes very large, the Kalman filter is really effective in filtering out the effect of the pose estimate given by R3 because of the large value of the variance of the pose estimate.

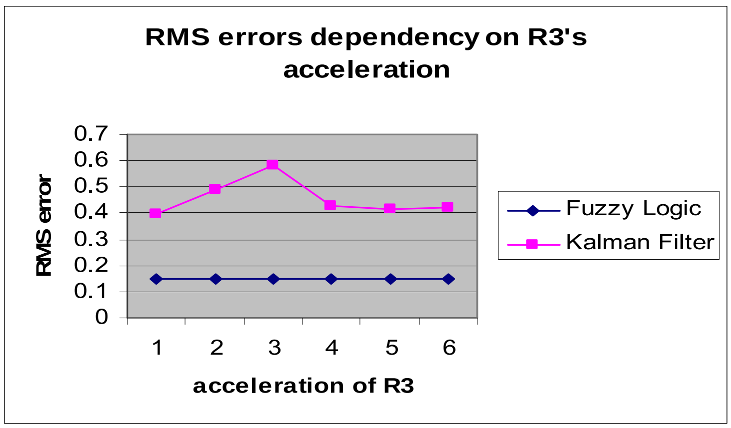


Fig. 23. The RMS error graphical comparison. The dependency of the RMS errors on R3's acceleration is compared and plotted.

CHAPTER VI

SUMMARY AND CONCLUSION

Summary

An approach of localization of multiple robots using fuzzy logic in a no-landmark scenario is developed. All the robots give pose estimates for all the other robots. The pose estimate by the one robot for another is based on its own pose value calculated using odometry sensor. The odometry sensors generally provide inaccurate estimates due to wheel slippages. The wheel slippage depends upon various physical factors like acceleration of the robot, the surface, the weight of the robot etc. But, given the surface and the weight of the robot to be constant, the prime influencing factor for the wheel slippage is the robot acceleration. A precise mathematical model of this dependency is almost practically impossible, so it is generally modeled as Gaussian distribution. Here, this uncertainty is modeled as a fuzzy membership trapezoidal function based on the acceleration of the robot. Also the estimate by one robot done by another robot also depends upon the distance of separation between the two robots, which can be measured by an omni-directional stereo camera setup. There is inherent uncertainty in this sensor because of the limited resolution. Therefore, if the distance of separation is large enough then the pose estimate is expected to be more inaccurate. This inaccuracy is also modeled as a similar fuzzy membership function. These two factors are combined together using fuzzy combination rule matrix to find the characteristic parameters to finally construct a combined fuzzy membership function. The fuzzy membership functions constructed in this way are collected from all the robots and are fused together to give a possibility distribution function for a pose parameter for each robot.

To fuse all the pose estimates, generally, a weighted average method is used often implemented in some form of Kalman filter technique. In the fuzzy logic approach, it is done using fuzzy combination operators to generate possibility distribution functions. These possibility distribution functions are then defuzzified using various defuzzification techniques like mean of max, centroid of area and cutoff centroid of area to give crisp values of pose parameters.

In addition to the fuzzy logic approach discussed above, a Kalman filter approach is also developed and then compared with it.

Conclusion

The fuzzy logic approach developed here presents some advantages over the Kalman filter techniques. The fuzzy logic approach is more robust than a Kalman filter to outlier robots, which provide way inaccurate pose estimates. Kalman filter gives less weight to an outlier pose estimate, whereas, a fuzzy logic approach eliminates it altogether. The basic Kalman filter approach is less computationally intensive compared to the fuzzy logic approach developed here.

REFERENCES

[1] S. Tsugawa, “Traveling Path Measurement for Navigation of Robot Rover by Precise Counting of Wheel Revolutions,” *ACTA IMEKO*, vol. 5, pp. 41-48, 1985.

[2] L. E. Parker, “Current State of the Art in the Distributed Autonomous Mobile Robotics,” in *Distributed Autonomous Robotic Systems,* L. E Parker, G. Bekey and J. Barhen, eds., Springer-Tokyo, Japan, 2000, vol.4, pp 3-12.

[3] I. M. Rekleitis, G. Dudek, E. Evangelos. Milios, “Multi-Robot Exploration of an Unknown Environment, Efficiently Reducing the Odometry Error,” in *Proc. Int. Joint Conf. on Artificial Intelligence (IJCAI)*, Nagoya Japan, Sept 1997, pp. 1340-1345.

[4] A. Howard, M. J. Mataric´ and G. S. Sukhatme, "Putting the `I' in `Team': An Ego-Centric Approach to Cooperative Localization", in *Proc. of the IEEE Int. Conf. on Robotics an Automation (ICRA03)*, Taipei, Taiwan, Sept. 2003, pp. 868—892.

[5] A. Howard, M.J. Mataric´ and G.S. Sukhatme, "Localization for Mobile Robot Teams: A Distributed MLE Approach", in *Experimental Robotics VIII,* Bruno Siciliano and Paulo Dario eds. Springer-Verlag, Berlin, Germany, 2003, pp.146-155.

[6] A. Howard, M.J. Mataric´ and G.S. Sukhatme, "Cooperative Relative Localization for Mobile Robot Teams: An Ego-centric Approach", in *Multi-Robot Systems: from Swarms to Intelligent Automata*, Alan C. Schultz and Lunee E. Parker and Frank E. Schneider eds., Kluwer, Los Angeles, CA, 2003, pp. 65-76.

[7] R. Madhavan, K. Fregene, and L. E. Parker, “Distributed Heterogeneous Outdoor Multirobot Localization,” in *Proc. of IEEE Int. Conf. on Robotics and Automation,* Arlington, VA , 2002, pp.

[8] J. Gluckman, S. K. Nayar, and K. J. Thoresz, "Real-time omnidirectional and panoramic stereo," in *Proc. DARPA Image Understanding Workshop*, Monterey, CA, Nov. 1998, pp. 299-303.

[9] D. Fox, W. Burgard, H. Kruppa, and S. Thrun, “Collaborative multirobot localization,” in *Proc. of the 23rd Annual German Conf. on Artificial Intelligence (KI)*, Bonn, Germany, Sep. 1999, pp. 255--266.

[10] “How Multiple Numbers of Humans and Robots Can Perform in Cooperation,” Artificial Intelligence Laboratory, Graduate School of Information Science, Nara Institute of Science and Technology (NAIST), Japan. http://www.incx.nec.co.jp/robot/univ/03/univ_e03.html. Accessed, Aug 2004.

[11] H. R. Everett, R. T. Laird, D. M. Carroll, G. Gilbreath, T. A. Heath-Pastore, R.S. Inderieden, T. Tran, K. Grant and D. M. Jaffee, "Multiple Resource Host Architecture (MRHA) for the Mobile Detection Assessment Response System (MDARS)," *Technical Document 3026, Revision A,* Space and Naval Warfare Systems Center, San Diego, CA, Sept. 2000.

[12] 2001 News Releases “NASA Bulldozer Rovers Could Get the Scoop on Mars.” http://www.jpl.nasa.gov/releases/2001/release_2001_210.html. Oct. 29, 2001.

[13] R. Colin Johnson, "Robots Readied to Take on Search-and-Rescue Duties." http://www.eetimes.com/at/news/showArticle.jhtml?articleID=19201885, Apr. 27, 2004.

[14] J. Clausen. "SARGE Robot is Latest Battlefiled Standard." http://www.sandia.gov/media/sarge.htm, Jan. 19, 1996.

[15] "Lawn Mowing for Lazybones." http://www.wired.com/news/technology/0,1282,62853,00.html?tw=wn_story_top5 Apr. 2004.

[16] J. Wawerla, G. S. Sukhatme and M. J. Mataric´. "Collective Construction with Multiple Robots", in *IEEE/RSJ International Conference on Robotics and Intelligent Systems* (*IROS*), Lausanne, Switzerland, Oct. 2002, pp. 2696 – 2701.

[17] Columbia Automated Vision Environment (CAVE) Lab at Columbia University, New York, NY, http://www1.cs.columbia.edu/CAVE/. Accessed, Aug. 2004.

[18] R. Chatila and J. Laumond, "Position Referencing and Consistent World Modeling for Mobile Robots," in *Proc. of IEEE Int. Conf. on Robotics and Automation,* St. Louis, MO, 1985, pp.138-145.

[19] J. L. Crowley, "Dynamic World Modeling for an Intelligent Mobile Robot Using a Rotating Ultra-Sonic Ranging Device," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, St. Louis, MO, 1985, pp. 128-135.

[20] Y. Watanabe and S. Yuta, "Estimation of Position and Its Uncertainty in the Dead Reckoning System of the Wheeled Mobile Robot," in *Proc. of 20th ISIR,* Tokyo, Japan, 1990, pp.205-212.

[21] T. Tsubouchi and S.Yuta, “Map Assisted Vision System of Mobile Robots for Reckoning in a Building Environment,” in *Proc. of IEEE Int. Conf. on Robotics and Automation*, Raleigh, NC, 1987, pp. 1978-1984.

[22] K. Kato, H. Ishiguro and M. Barth, “Identifying and Localizing Robots in a Multi-Robot System Environment,” in *IEEE/RSJ Int. Conf. on Robots & Systems*, Kyungju, South Korea, 1999, vol. 2, pp. 966-971.

[23] L. Matthies and S.A. Shafer, “Error Modeling in Stereo Navigation,” *IEEE Journal of Robotics and Automation*, 1987, vol. RA-3, no. 3, pp. 239-248.

[24] H. Nasr and B. Bhanu, “Landmark Recognition for Autonomous Mobile Robots,” in *Proc. of IEEE Int. Conf. on Robotics and Automation*, Philadelphia, PA, 1988, pp. 1218-1223.

[25] E. Krotkov, “Mobile Robot Localization Using a Single Image,” in *Proc. IEEE Int. Conf. on Robotics and Automation*, Scottsdale, AZ, 1989, pp. 978-983.

[26] P. Cánovas, K. LeBlanc, and A. Saffiotti, “Cooperative Object Localization Using Fuzzy Logic,” in *Proc. of the IEEE Int. Conf. on Methods and Models in Automation and Robotics.* (MMAR) Miedzyzdroje, Poland, 2003, pp. 773-778.

[27] J. S. Gutmann and D. Fox, “An Experimental Comparison of Localization Methods Continued,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'02)*, Lausanne, Switzerland, October 2002

[28] A. Howard, M. J. Mataric´ and G. S. Sukhatme, "Localization for Mobile Robot Teams Using Maximum Likelihood Estimation," in *IEEE/RSJ International*

Conference on Robotics and Intelligent Systems, Lausanne, Switzerland, Oct. 2002, pp. 434-459.

[29] R. Kurazume, S. Nagata and S. Hirose, “Cooperative Positioning with Multiple Robots,” in *Proc. IEEE Int. Conf. on Robotics and Automation,* San Diego ,CA, 1994, vol. 2, pp. 1250-1257.

[30] N. Bulusu, J. Heidemann, and D. Estrin, “GPS-less Low Cost Outdoor Localization for Very Small Devices,” *IEEE Personal Comm. Magazine*, Oct. 2000, vol. 7 no. 5, pp. 28-34.

[31] A. Ward, A. Jones and A. Hopper, “A New Location Technique for the Active Office,” *IEEE Personal Comm.,* Oct. 1997, vol. 4, no. 5, pp. 42- 47.

[32] S.I. Roumeliotis and G.A. Bekey, “Collective Localization: A Distributed Kalman Filter Approach to Localization of Groups of Mobile Robots,” in *Proc. 2000 IEEE International Conference on Robotics and Automation*, San Francisco, CA, Apr. 2000, pp. 2958-2965.

[33] S.I. Roumeliotis and G.A. Bekey, "Bayesian Estimation and Kalman Filtering: A Unified Framework for Mobile Robot Localization," in *Proc. 2000 IEEE Int. Conf. on Robotics and Automation*, San Francisco, CA, Apr. 2000, pp. 2985-2992.

[34] S. I. Roumeliotis and I. M. Rekleitis, “Analysis of Multi Robot Localization Uncertainty Propagation,” in *Proc. 2003 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems,* LasVegas, NV, Oct. 2003, pp. 1763-1770.

[35] M. Dietl, J. S. Gutmann, and B. Nebel, “Cooperative Sensing in Dynamic Environments,” in *Proc. of IROS 2001*, Maui, HI, Oct. 2001, pp. 1706-1713.

[36] P. Buschka, A. Saffiotti, and Z. Wasik, "Fuzzy Landmark-Based Localization for a Legged Robot," in *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems* (IROS), Takamatsu, Japan, 2000, pp. 1205-1210.

[37] A. Saffiotti and L. P. Wesley, "Perception-based Self-localization Using Fuzzy Locations," in *Reasoning with Uncertainty in Robotics*, M. van Lambalgen L. Dorst and F. Voorbraak, eds. in LNAI, Springer-Verlag, Berlin, Germany, 1996, pp. 368-385.

[38] A. Saffiotti, "Fuzzy Logic in Autonomous Robotics: Behavior Coordination," in *Proc. of the 6th IEEE Int. Conf. on Fuzzy Systems*, Barcelona, Spain, 1997, pp. 573-578.

[39] D. Herrero Pérez, H. Martínez Barberá and A. Saffiotti, "Fuzzy Self-Localization using Natural Features in the Four-Legged League," in *Proc. of the Int. RoboCup Symposium*. Lisbon, Portugal, 2004.

[40] Greg Welch and Gary Bishop. "An Introduction to the Kalman Filter, SIGGRAPH 2001 course 8," in *Computer Graphics, Annual Conference on Computer Graphics & Interactive Techniques*. ACM Press, Addison-Wesley, Los Angeles, CA, USA, SIGGRAPH 2001 course pack edition, 2001.

[41] R. R. Negenborn, "Robot Localization and Kalman Filters," master's thesis INF/SCR-0309. Utrecht University, Utrecht, Netherlands, 2003.

[42] A. Stroupe, M. C. Martin, and T. Balch, "Merging Gaussian Distributions for Object Localization in Multi-Robot Systems," in *Proc. of the ISER 2000 7th International Symposium on Experimental Robotics*, Honolulu, HI, Dec. 2000.

[43] Basics of Fuzzy Logic and Fuzzy Set Theory.
http://www.ncst.ernet.in/kbcs/vivek/issues/11.1/sam/sam.html. Accessed, Aug 2004.

[44] A. D. Marshall and R. R. Martin, "Computer Vision, Models and Inspection," World Scientific Publishing Co.
http://www.cs.cf.ac.uk/Dave/Vision_lecture/node11.html. Accessed, Aug. 2004.

[45] Discussion (Question & Answer) Perception-Based Self Localization using Fuzzy Locations. http://www.pa.info.mie-u.ac.jp/bioele/wsc1/papers/d050.html. Accessed, Aug. 2004.

VITA

Dhirendra Kumar Sinha received his Bachelor of Technology degree in mechanical engineering from Indian Institute of Technology, Guwahati, India in May 2001. From Fall 2002 onwards, he attended the Master of Science Program in mechanical engineering, Texas A&M University, College Station and received his Master of Science degree in December 2004.

Permanent mailing address:

Plot # 91, Brindavan Colony

Bolarum, Secunderabad

India